



## SDK интеграции (HTTP API)

Last update 12/27/2018

# Содержание

<b>1</b>	<b>Общие соглашения</b> .....	<b>6</b>
<b>2</b>	<b>Инфраструктура</b> .....	<b>7</b>
2.1	Серверное HTTP API .....	7
2.1.1	Получение уникального идентификатора.....	7
2.1.2	Получение списка серверов .....	7
2.1.2.1	Список всех серверов в домене .....	7
2.1.2.2	Информация о конкретном сервере.....	7
2.1.3	Получение списка источников видео (камер).....	7
2.1.4	Получение живого потока от видеокамеры .....	9
2.1.4.1	Общие сведения.....	9
2.1.4.2	Видео по протоколу HLS.....	10
2.1.4.3	Видео по протоколу RTSP.....	10
2.1.4.4	Видео по протоколу HTTP .....	10
2.1.4.5	Туннелирование RTSP по HTTP.....	11
2.1.4.6	Видео в формате H.264 .....	11
2.1.4.7	Получение потоков высокого и низкого качества .....	11
2.1.4.8	Настройка туннелирования RTSP по HTTP в VLC .....	11
2.1.5	Получение снимка с видеокамеры .....	12
2.1.6	Получение содержания архивов .....	12
2.1.6.1	Получение списка архивов, в которые ведется запись: .....	12
2.1.6.2	Получение содержания архива по умолчанию:.....	13
2.1.6.3	Получение содержания конкретного архива: .....	13
2.1.7	Получение информации об архиве .....	14
2.1.7.1	Глубина архива .....	14
2.1.7.2	Объем записи в архив по указанной видеокамере .....	14
2.1.8	Получение информации о повреждении архива.....	14
2.1.9	Получение архивного потока.....	15
2.1.9.1	Получение архивного потока из архива по умолчанию.....	16
2.1.9.2	Получение архивного потока из конкретного архива.....	17
2.1.9.3	Архивное видео по протоколу RTSP .....	17
2.1.9.4	Архивное видео по протоколу HTTP.....	17
2.1.9.5	Туннелирование RTSP по HTTP.....	17
2.1.9.6	Архивное видео в формате H.264.....	17

2.1.9.7	Получение информации об архивном потоке.....	17
2.1.9.8	Управление архивным потоком .....	18
2.1.9.9	Покадровый просмотр архива .....	18
2.1.9.9.1	Получение кадра по моменту времени .....	18
2.1.9.9.2	Получение времени регистрации кадра.....	18
2.1.10	Получение подписанных ссылок на видео потоки.....	18
2.1.11	Получение комментариев из архива.....	19
2.1.12	Поиск в архиве .....	20
2.1.12.1	Общий интерфейс.....	20
2.1.12.1.1	Запрос на поиск .....	20
2.1.12.1.1.1	Поиск по одному источнику .....	20
2.1.12.1.1.2	Поиск по нескольким источникам .....	20
2.1.12.1.1.3	Результат .....	20
2.1.12.1.2	Запрос результатов поиска.....	21
2.1.12.1.3	Завершение поиска .....	21
2.1.12.2	API Поиск по лицам.....	21
2.1.12.3	API Поиск лиц "свой"- "чужой" .....	22
2.1.12.4	Определение признака "свой"- "чужой" по изображению .....	23
2.1.12.5	API Поиск по номерам.....	24
2.1.12.6	API Интеллектуальный поиск MomentQuest (VMDA) .....	25
2.1.12.6.1	Типы запросов и их параметры .....	27
2.1.12.6.1.1	Пребывание объекта в зоне (queryType="zone") .....	27
2.1.12.6.1.2	Перемещение объекта из одной зоны в другую (queryType="transition").....	28
2.1.12.6.1.3	Пересечение линии (queryType="line").....	28
2.1.12.6.2	Дополнительные условия .....	29
2.1.12.6.2.1	Тип объекта (objectProperties/category) .....	30
2.1.12.6.2.2	Размер объекта (objectProperties/size).....	31
2.1.12.6.2.3	Цвет объекта (objectProperties/color) .....	32
2.1.12.6.2.4	Скорость (conditions/velocity).....	33
2.1.12.6.2.5	Направление движения (conditions/directions).....	34
2.1.12.6.2.6	Длительность (conditions/duration) .....	35
2.1.12.6.2.7	Количество объектов (condtions/count) .....	35
2.1.13	Получение списка групп и их содержимого .....	36
2.1.13.1	Получение списка всех доступных групп .....	36
2.1.13.2	Получение содержимого группы .....	37
2.1.13.3	Получение списка групп, содержащих указанную видеокамеру.....	37

2.1.14	Экспорт .....	38
2.1.14.1	Запуск экспорта .....	38
2.1.14.2	Запрос статуса экспорта .....	39
2.1.14.3	Завершение экспорта.....	40
2.1.14.4	Скачивание файла.....	40
2.1.15	Получение списка детекторов видеокамеры .....	40
2.1.16	Получение информации о срабатках детекторов и тревогах.....	41
2.1.16.1	Получение списка тревог .....	41
2.1.16.2	Получение списка срабатываний детекторов.....	42
2.1.17	Управление телеметрией.....	44
2.1.17.1	Получение списка устройств телеметрии для источника видео.....	44
2.1.17.2	Захват сессии управления телеметрией.....	44
2.1.17.3	Поддержание актуальности сессии.....	45
2.1.17.4	Освобождение сессии.....	45
2.1.17.5	Управление степенями свободы .....	45
2.1.17.5.1	Получение информации о степенях свободы.....	45
2.1.17.5.2	Изменение наклона и поворота.....	46
2.1.17.5.3	Изменение одной из степеней свободы.....	46
2.1.17.5.4	Переход по точке на экране .....	46
2.1.17.5.5	Увеличение области изображения.....	47
2.1.17.5.6	Автоматическая фокусировка и диафрагма .....	47
2.1.17.6	Управление предустановками (preset).....	47
2.1.17.6.1	Получение списка предустановками.....	47
2.1.17.6.2	Создание и изменение предустановки .....	48
2.1.17.6.3	Переход к предустановке и удаление .....	48
2.1.17.7	Получение информации об ошибках .....	48
2.1.18	Работа с макрокомандами .....	48
2.1.19	Получение статистики.....	49
2.1.20	Получение данных из системного журнала.....	49
2.1.21	Получение информации о загрузке Серверов.....	51
2.1.22	Получение информации об установленной версии Сервера.....	52
2.1.23	Переключение состояния виртуального IP-устройства (HttpListener) .....	52
2.2	Клиентское HTTP API.....	53
2.2.1	Запросы для работы с раскладками и видеостенами .....	53
2.2.1.1	Порядок работы.....	53

2.2.1.2	Получение списка раскладок .....	53
2.2.1.3	Переключение раскладки на экране .....	54
2.2.1.4	Получение списка видеокамер, отображаемых на раскладке в данный момент .....	54
2.2.1.5	Удаление и добавление видеокамер .....	55
2.2.1.6	Получение списка мониторов .....	55
2.2.1.7	Выбор активного монитора.....	55
2.2.1.8	Перевод видеокамеры в режим архива .....	56
2.2.1.9	Переход на раскладку с видеокамерой в режиме погружения .....	56
2.2.1.10	Переход на раскладку с видеокамерой в режиме интеллектуального поиска .....	56
2.2.1.11	Переход к сохраненным результатам интеллектуального поиска в архиве .....	56

# 1 Общие соглашения

HTTP сервер NGP возвращает результаты вызова методов в формате JSON.

**Примечание**  
[Инструкция по настройке Web-сервера.](#)  
Порт Web-сервера по умолчанию - **80**, префикс - / (пустой)

**Примечание**  
При открытии запроса в браузере GET не используется

Для выполнения запросов необходима авторизация. Поддерживаемый тип авторизации - basic.

Данные пользователя необходимо добавлять во все HTTP запросы в следующем виде:

```
http://[имя_пользователя]:[пароль]@[IP-адрес]:[порт]/[префикс]
```

POST-запросы должны содержать тело в формате JSON.

Существует ограничение на количество активных запросов и запросов, находящихся в очереди.

При переполнении очереди запросов будет возвращена ошибка с кодом **503** - Search query rejected. Too many requests.

## 2 Инфраструктура

### 2.1 Серверное HTTP API

#### 2.1.1 Получение уникального идентификатора

GET http://IP-адрес:порт/префикс/uuid - на каждый запрос генерирует уникальный идентификатор (UUID).

Уникальный идентификатор может понадобиться, например, для получения информации о последнем отображенном кадре архивного видео или для управления архивным потоком.

**Пример ответа:**

```
{
  "uuid": "2736652d-af5f-4107-a772-a9d78dfaa27e"
}
```

#### 2.1.2 Получение списка серверов

##### 2.1.2.1 Список всех серверов в домене

GET http://IP-адрес:порт/префикс/hosts/ - получить список всех хостов в домене.

**Пример ответа:**

```
[ "SERVER1", "SERVER2" ]
```

##### 2.1.2.2 Информация о конкретном сервере

GET http://IP-адрес:порт/префикс/hosts/HOSTNAME - получить информацию о конкретном хосте.

**Пример ответа:**

```
{
  "hostname": "SERVER2",
  "domainInfo": {
    "domainName": "DomainName",
    "domainFriendlyName": "Пользовательское название домена, если есть"
  },
  "platformInfo": {
    "machine": "ARM9",
    "os": "Linux",
    "licenseStatus": "Expired",
    "timeZone": "+180" // GMT+3
  }
}
```

#### 2.1.3 Получение списка источников видео (камер)



Аххон Next 4.1.1.7773 и старше.  
До этого см. [Получение списка источников видео \(камер\)](#)

GET http://IP-адрес:порт/префикс/camera/list - получить все доступные оригинальные источники (камеры) Аххон-домена. Возвращаемые идентификаторы VIDEOSOURCEID будут иметь вид "HOSTNAME/ObjectTypeId/Endpoint.Name". Также будут возвращены friendly name и прочая метаинформация.

возможные параметры:

**limit** - определяет максимальное количество возвращаемых результатов, по умолчанию 1000.

**next\_page** - значение параметра **nextPageToken**, который будет в ответе, если запрос вернул не все результаты. Применяется для получения следующих значений.

**filter** - параметр, который позволяет получить подмножество результатов в соответствии с фильтром. На данный момент может принимать значения **HOSTNAME** или **VIDEOSOURCEID**.

**⚠ Внимание!**

Если имя сервера Endpoint, то запрос GET `http://IP-адрес:порт/префикс/camera/list?filter=Endpoint` вернет все видеокамеры.

Пример ответа:

```
{
  "cameras": [
    {
      "archives": [
        {
          "accessPoint": "hosts/QA-T40/DeviceIpint.1/SourceEndpoint.video:0:0",
          "isEmbedded": false,
          "storage": "hosts/QA-T40/MultimediaStorage.AliceBlue/MultimediaStorage",
          "storageDisplayName": "Архив AliceBlue"
        }
      ],
      "audioStreams": [
        {
          "accessPoint": "hosts/QA-T40/DeviceIpint.1/SourceEndpoint.audio:0"
        }
      ],
      "detectors": [
        {
          "accessPoint": "hosts/QA-T40/AVDetector.1/EventSupplier",
          "displayName": ""
        }
      ],
      "displayName": "Камера",
      "ipAddress": "172.17.11.22",
      "offlineDetectors": [],
      "ptzs": [],
      "videoStreams": [
        {
          "accessPoint": "hosts/QA-T40/DeviceIpint.1/SourceEndpoint.video:0:0"
        },
        {
          "accessPoint": "hosts/QA-T40/DeviceIpint.1/SourceEndpoint.video:0:1"
        }
      ]
    }
  ],
  {
```



## 2.1.4 Получение живого потока от видеокамеры

### На странице:

- [Общие сведения](#)
- [Видео по протоколу HLS](#)
- [Видео по протоколу RTSP](#)
- [Видео по протоколу HTTP](#)
- [Туннелирование RTSP по HTTP](#)
- [Видео в формате H.264](#)

[Получение подписанных ссылок на видео потоки](#)

### **Внимание!**

Получение звука возможно только с 64-битного Сервера.

### 2.1.4.1

#### Общие сведения

GET <http://IP-адрес:порт/префикс/live/media/VIDEOSOURCEID?параметры>.

где **VIDEOSOURCEID** - трехкомпонентный идентификатор endpoint-а источника формата (см. [Получение списка источников видео \(камер\)](#)). Например, "SERVER1/DeviceIpint.3/SourceEndpoint.video:0:0".

Параметры:

- **format** - "rtsp", "hls".

По протоколам RTSP и HLS видео можно получать в исходном формате без рекомпрессии. При этом протокол HLS поддерживает только формат H.264.

Во всех других случаях сервер производит рекомпрессию в формат MJPEG.

### **Внимание!**

Если запрашивается видео в формате, отличном от исходного, то будет произведена рекомпрессия, что приведет к увеличению загрузки Сервера.

- **w** – значение ширины кадра
- **h** – значение высоты кадра.

### **Примечание**

Если значения параметров **h** и **w** больше размеров оригинального видео, видео будет получено с оригинальными размерами.

Уменьшение ширины и высоты возможно только дискретно - в 2, 4, 8 и т. д. раз. Если указаны размеры, не соответствующие 1/2, 1/4 и т. д. размеров оригинального видео, то видео будет получено с размерами, кратными размерам оригинального видео, наиболее близкими к указанным.

- **fr** - значение частоты кадров.

### **Внимание!**

Данный параметр актуален только для видео в формате MJPEG.

**Пример запроса:**

```
GET http://IP-адрес:порт/префикс/live/media/HOSTNAME/DeviceIpint.23/SourceEndpoint.video:0:0?w=640&h=480
```

**2.1.4.2 Видео по протоколу HLS**

Видео по протоколу HLS можно получать только в исходном формате. При получении видео по протоколу HLS также используются следующие параметры:

**keep\_alive** - время в секундах, через которое необходимо продлевать поток.

**hls\_time** - длина сегмента в секундах.

**hls\_list\_size** - максимальное количество записей списка воспроизведения. Если задано значение **0**, список будет содержать все сегменты.

**hls\_wrap** - порядковый номер сегмента, после которого счётчик обнулится. Если задано значение **0**, то счётчик не обнуляется.

**Пример запроса:**

```
GET http://IP-адрес:порт/префикс/live/media/HOSTNAME/DeviceIpint.23/SourceEndpoint.video:0:0?format=hls&keep_alive=60
```

**Пример ответа:**

```
{
  "keep_alive_seconds": 60,
  "keep_alive_url": "/live/media/hls/keep?stream_id=7e9d8c93-80e2-4521-9a54-cb854fe3cd2d",
  "stop_url": "/live/media/hls/stop?stream_id=7e9d8c93-80e2-4521-9a54-cb854fe3cd2d",
  "stream_url": "/hls/7e9d8c93-80e2-4521-9a54-cb854fe3cd2d/playlist.m3u8"
}
```

где **keep\_alive\_url** - ссылка для продления жизни потока;

**stop\_url** - ссылка для остановки видеопотока;

**stream\_url** - ссылка по-которой будет доступен список сегментов.

**⚠ Внимание!**

Видео по протоколу HLS будет доступно через несколько секунд после ответа

Для проигрывания видео по протоколу HLS необходимо использовать параметр **stream\_url** из ответа следующим образом:

```
ffplay "http://root:root@10.0.12.65:80/hls/c83b48d5-2ab7-49eb-91ef-593f808d4e51/playlist.m3u8"
```

**2.1.4.3 Видео по протоколу RTSP**

Видео по протоколу RTSP передается только в оригинальном формате.

```
GET rtsp://логин:пароль@IP-адрес:554/hosts/HOSTNAME/DeviceIpint.23/SourceEndpoint.video:0:0
```

**2.1.4.4 Видео по протоколу HTTP**

```
GET ffplay.exe -v debug "http://логин:пароль@IP-адрес:8001/asisp-api/live/media/HOSTNAME/DeviceIpint.23/SourceEndpoint.video:0:0?w=1600&h=0"
```

**⚠ Внимание!**

HTTP передает видео только в формате mjpeg, задание параметров **w** и **h** обязательно.

### 2.1.4.5 Туннелирование RTSP по HTTP

см. [Настройка туннелирование RTSP по HTTP в VLC](#)

Видео через туннель передается только в оригинальном формате.

Примеры:

```
GET ffmpeg -rtsp_transport http "rtsp://логин:пароль@IP-адрес:80/rtspproxy/hosts/HOSTNAME/DeviceIpint.23/SourceEndpoint.video:0:0"
```

```
GET Для VLC: rtsp://логин:пароль@IP-адрес:80/rtspproxy/hosts/HOSTNAME/DeviceIpint.23/SourceEndpoint.video:0:0
```

### 2.1.4.6 Видео в формате H.264

Для получения живого видео в оригинальном формате H.264 необходимо использовать RTSP или туннель RTSP по HTTP.

### 2.1.4.7 Получение потоков высокого и низкого качества

- ✓ [Получение списка источников видео \(камер\)](#)
- [Получение живого потока от видекамеры](#)

Общий случай:

- GET http://IP-адрес:порт/префикс/live/media/SERVER1/DeviceIpint.3/SourceEndpoint.video:0:0?w=1600&h=0 - поток высокого качества
- GET http://IP-адрес:порт/префикс/live/media/SERVER1/DeviceIpint.3/SourceEndpoint.video:0:1?w=1600&h=0 - поток низкого качества

RTSP:

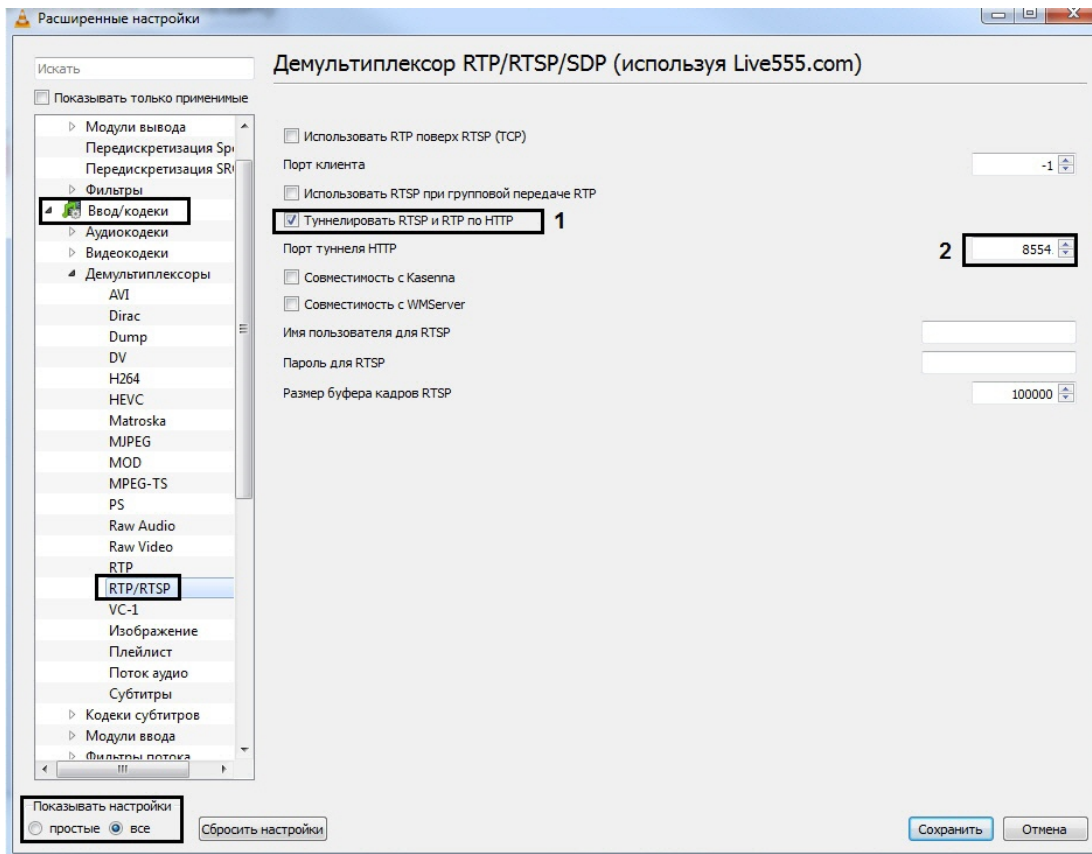
- GET rtsp://логин:пароль@IP-адрес:554/hosts/SERVER1/DeviceIpint.3/SourceEndpoint.video:0:0 - поток высокого качества
- GET rtsp://логин:пароль@IP-адрес:554/hosts/SERVER1/DeviceIpint.3/SourceEndpoint.video:0:1 - поток низкого качества

Туннелирование RTSP по HTTP:

- GET rtsp://логин:пароль@IP-адрес:80/rtspproxy/hosts/SERVER1/DeviceIpint.3/SourceEndpoint.video:0:0 - поток высокого качества
- GET rtsp://логин:пароль@IP-адрес:80/rtspproxy/hosts/SERVER1/DeviceIpint.3/SourceEndpoint.video:0:1 - поток низкого качества

### 2.1.4.8 Настройка туннелирования RTSP по HTTP в VLC

Для настройки туннелирования в VLC необходимо установить флажок **Туннелировать RTSP и RTP по HTTP (1)**, и указать порт web-сервера (**2**, см. [Настройка Web-сервера](#)).



## 2.1.5 Получение снимка с видеокамеры

GET `http://IP-адрес:порт/префикс/live/media/snapshot/VIDEOSOURCEID?параметры`.

где **VIDEOSOURCEID** - трехкомпонентный идентификатор endpoint-а источника (см. [Получение списка источников видео \(камер\)](#)).

Параметры:

**w** – значение ширины кадра.

**h** – значение высоты кадра.

### **Примечание**

По умолчанию период обновления кадра 30 секунд. Для его изменения необходимо создать системную переменную `NGP_SNAPSHOT_TIMEOUT` и задать необходимое значение в миллисекундах.

### Пример запроса:

Получение снимка в оригинальном разрешении: GET `http://IP-адрес:порт/префикс/live/media/snapshot/HOSTNAME/DeviceIpint.23/SourceEndpoint.video:0:0`

Получение снимка в разрешении 640\*480: GET `http://IP-адрес:порт/префикс/live/media/snapshot/HOSTNAME/DeviceIpint.23/SourceEndpoint.video:0:0?w=640&h=480`

## 2.1.6 Получение содержания архивов

### 2.1.6.1 Получение списка архивов, в которые ведется запись:

GET `http://P-адрес:порт/префикс/archive/list/VIDEOSOURCEID`

где **VIDEOSOURCEID** - трехкомпонентный идентификатор endpoint-а источника формата (см. [Получение списка источников видео \(камер\)](#)). Например, "SERVER1/DeviceIpint.3/SourceEndpoint.video:0:0".

### Пример ответа:

```

{
  "archives" : [
    {
      "default" : true,
      "name" : "hosts/SERVER1/MultimediaStorage.AliceBlue/MultimediaStorage"
    },
    {
      "default" : false,
      "name" : "hosts/SERVER1/MultimediaStorage.AntiqueWhite/MultimediaStorage"
    }
  ]
}

```

### 2.1.6.2 Получение содержания архива по умолчанию:

GET http://IP-адрес:порт/префикс/archive/contents/intervals/VIDEOSOURCEID/ENDTIME/BEGINTIME?  
limit=COUNT&scale=SIZE - получение содержания архива, начиная от момента времени BEGINTIME, заканчивая  
моментом времени ENDTIME.

В случае, если BEGINTIME не указан, подразумевается значение, соответствующее бесконечному будущему. Если при этом ENDTIME также не указан, подразумевается его значение, соответствующее бесконечному прошлому. Вместо BEGINTIME и ENDTIME могут быть использованы слова "past" или "future", означающие бесконечное прошлое и бесконечное будущее соответственно.

Необязательный параметр limit указывает, каким количеством интервалов следует ограничиться. Значение limit по умолчанию равно **100**.

Необязательный параметр scale указывает, при каком минимальном временном расстоянии между двумя интервалами они будут выданы как два различных интервала (а не склеены в один). Значение scale по умолчанию равно **0**.

Порядок выдачи интервалов соответствует отношению между заданными BEGINTIME и ENDTIME (по возрастанию, если BEGINTIME<ENDTIME, и по убыванию, если ENDTIME<BEGINTIME). При этом начало и конец интервала всегда выдаются в естественном порядке, т.е. значение времени начала интервала будет меньше либо равно времени конца интервала).

В возвращаемом ответе (json объекте) массив, содержащий интервалы, помещается в свойство с именем **intervals**.

В возвращаемом ответе (json объекте) отдельное свойство **more** - булевское значение, сигнализирующее о том, был ли выбран указанный в запросе отрезок времени полностью (false), или остались интервалы, которые не были возвращены из-за достижения предельного числа возвращаемых интервалов (true).

### 2.1.6.3 Получение содержания конкретного архива:

GET http://IP-адрес:порт/префикс/archive/contents/intervals//VIDEOSOURCEID/future/past?archive=Имя\_Архива

#### Пример запроса:

GET http://IP-адрес:порт/префикс/archive/contents/intervals/SERVER1/Devicelpint.2/SourceEndpoint.video:  
0:0/20101230T103904.000/20101230T103959.000?limit=3

#### Пример ответа:

```

{
  "intervals" :
  [
    { begin: "20101230T103950.000", end: "20101230T103955.230" },
    { begin: "20101230T103923.110", end: "20101230T103941.870" }
  ],

```

```
"more" : true
}
```

**Примечание**  
Время возвращается в формате UTC

## 2.1.7 Получение информации об архиве

### 2.1.7.1 Глубина архива

GET http://IP-адрес:порт/префикс/archive/statistics/depth/VIDEOSOURCEID/ENDTIME/BEGINTIME?threshold=7 - получение информации о глубине архива, начиная от момента времени BEGINTIME, заканчивая моментом времени ENDTIME.

VIDEOSOURCEID - трехкомпонентный идентификатор endpoint-а источника (HOSTNAME/ObjectType.Id/Endpoint.Name).

**threshold** - не обязательный параметр, позволяющий задать пороговое значение (в днях), при превышении которого процедура склеивания интервалов будет закончена. Значение по умолчанию - 1 день.

**Примечание**  
Синтаксис ENDTIME и BEGINTIME описан в разделе [Получение содержания архивов](#)

#### Пример запроса:

```
GET http://localhost:80/archive/statistics/depth/SERVER1/DeviceIpint.23/SourceEndpoint.video:0:0?threshold=2
```

#### Пример ответа:

```
{
  "start": "20160823T141333.778000"
  ,"end": "20160824T065142"
}
```

где 20160823T141333.778000 - 20160824T065142 - интервал времени, для которого записи в архиве доступны.

### 2.1.7.2 Объем записи в архив по указанной видеокамере

GET http://IP-адрес:порт/префикс/archive/statistics/capacity/VIDEOSOURCEID/ENDTIME/BEGINTIME - получение информации об объеме записи в архив указанной видеокамеры, начиная от момента времени BEGINTIME, заканчивая моментом времени ENDTIME.

**Примечание**  
Синтаксис ENDTIME и BEGINTIME описан в разделе [Получение содержания архивов](#)

#### Пример запроса:

```
GET http://IP-адрес:порт/префикс/archive/statistics/capacity/SERVER1/DeviceIpint.23/SourceEndpoint.video:0:0/past/future
```

#### Пример ответа:

```
{
  "size": 520093696
  ,"duration": 32345
}
```

где **size** - объем архива в байтах за указанный период;

**duration** - длительность архива в секундах за указанный период.

## 2.1.8 Получение информации о повреждении архива

```
GET http://IP-адрес:порт/префикс/archive/health/SERVER/ENDTIME/BEGINTIME?archive="name"&health=0
```

где,

- **SERVER** - имя Сервера.
- **ENDTIME** и **BEGINTIME** задает интервал проверки архива (см. [Получение содержания архивов](#)).
- **archive="name"** - имя архива из запроса получения списка архивов (см. [Получение содержания архивов](#)).
- **health=0** - есть повреждения архива, **health=1** - повреждений архива нет.

**⚠ Внимание!**  
Если в запросе нет параметра **archive** или **health**, то ответ будет содержать все значения данных параметров.

Пример запроса:

GET http://127.0.0.1:80/archive/health/SERVER/past/future?health=0

Пример ответа:

```
{
  "events" : [
    {
      "data" : {
        "archive" : "D:/archiveAntiqueWhite.afs",
        "health" : 0
      },
      "timestamp" : "20180907T101637.361014"
    },
    {
      "data" : {
        "archive" : "D:/archiveAntiqueWhite.afs",
        "health" : 0
      },
      "timestamp" : "20180907T102726.750134"
    }
  ]
}
```

где,

**timestamp** - время повреждения архива (UTC +0).

## 2.1.9 Получение архивного потока

### На странице:

- [Получение архивного потока из архива по умолчанию](#)
- [Получение архивного потока из конкретного архива](#)
- [Архивное видео по протоколу RTSP](#)
- [Архивное видео по протоколу HTTP](#)
- [Туннелирование RTSP по HTTP](#)
- [Архивное видео в формате H.264](#)

## Получение подписанных ссылок на видео потока

**⚠ Внимание!**  
Получение звука возможно только с 64-битного Сервера.

### 2.1.9.1 Получение архивного потока из архива по умолчанию

GET http://IP-адрес:порт/префикс/archive/media/VIDEOSOURCEID/STARTTIME?параметры,  
где

- **VIDEOSOURCEID** - трехкомпонентный идентификатор endpoint-а источника формата (см. [Получение списка источников видео \(камер\)](#)). Например, "SERVER1/DeviceIpint.3/SourceEndpoint.video:0:0".
- **STARTTIME** - время в формате ISO.

**⚠ Внимание!**  
Время задается в часовом поясе UTC+0

Параметры:

**speed** - скорость воспроизведения, может принимать отрицательные значения.

**format** - параметр может принимать значения "mjpeg", "rtsp" либо "hls". В случае, если формат не указан, указан "rtsp" или не распознан, сервер выбирает нативный формат (чтобы избежать перекодирования). Если при этом нативный формат не поддерживается клиентом, сервер выбирает WebM.

В случае, если не задано значение ни одного из двух вышеперечисленных параметров, скорость считается равной 0, а формат - JPEG, и запрос интерпретируется как запрос на кадровый просмотр архива.

**id** - уникальный идентификатор архивного потока (может отсутствовать). Необходим для получения информации о потоке или для управления им.

**w** - значение ширины кадра.

**h** - значение высоты кадра.

**fr** - значение частоты кадров.

**⚠ Внимание!**  
Данный параметр актуален только для видео в формате MJPEG.

**Пример запроса:**

```
GET http://IP-адрес:порт/префикс/archive/media/HOSTNAME/DeviceIpint.23/SourceEndpoint.video:0:0/20110608T060141.375?format=rtsp&speed=1&w=640&h=480
```

Для последующего получения информации о потоке необходимо обязательно присвоить потоку id.

```
http://IP-адрес:порт/префикс/archive/media/VIDEOSOURCEID/STARTTIME/20140723T120000.000?format=rtsp&speed=1&w=640&h=480&id=f03c6ccf-b181-4844-b09c-9a19e6920fd3
```

В качестве id можно использовать и другие значения, состоящие из букв и цифр латинского алфавита.

Рекомендуется для получения уникальных значений использовать функцию uuid (см. [Получение уникального идентификатора](#)).

**⚠ Внимание!**  
Архивное видео по протоколу HLS будет доступно в течение 30 секунд после ответа

Пример ответа:



```

{"http":
{"description":"RTP/RTSP/HTTP/TCP","path":"archive/hosts/HOSTNAME/DeviceIpint.1/
SourceEndpoint.video:0:0/20161206T060141.375000?speed=1&id=729955cd-7787-4d6c-87eb-
cd6dd6d4a940","port":"8554"}
,"rtsp":
{"description":"RTP/UDP or RTP/RTSP/TCP","path":"archive/hosts/HOSTNAME/DeviceIpint.1/
SourceEndpoint.video:0:0/20161206T060141.375000?speed=1&id=729955cd-7787-4d6c-87eb-
cd6dd6d4a940","port":"554"}
}

```

### 2.1.9.2 Получение архивного потока из конкретного архива

GET http://IP-адрес:порт/префикс/archive/media/VIDEOSOURCEID/STARTTIME?параметры&archive=hosts/SERVER1/MultimediaStorage.Имя\_Архива/MultimediaStorage

### 2.1.9.3 Архивное видео по протоколу RTSP

GET rtsp://логин:пароль@IP-адрес:554/archive/hosts/SERVER1/DeviceIpint.0/SourceEndpoint.video:0:0/20160907T050548.723000?speed=1

**speed** - обязательный параметр.

### 2.1.9.4 Архивное видео по протоколу HTTP

GET ffmpeg.exe -v debug "http://логин:пароль@IP-адрес:80/asip-api/archive/media/SERVER1/DeviceIpint.4/SourceEndpoint.video:0:0/20170112T113526?w=1600&h=0&speed=1"

### 2.1.9.5 Туннелирование RTSP по HTTP

см. [Настройка туннелирования RTSP по HTTP в VLC](#).

GET ffmpeg -rtsp\_transport http "rtsp://логин:пароль@IP-адрес:8554/rtspproxy/archive/hosts/SERVER1/DeviceIpint.4/SourceEndpoint.video:0:0/20170115T113526"

Для VLC: GET rtsp://логин:пароль@IP-адрес:8554/rtspproxy/archive/hosts/SERVER1/DeviceIpint.4/SourceEndpoint.video:0:0/20170115T113526

### 2.1.9.6 Архивное видео в формате H.264

Архивное видео в формате H.264 можно получить используя протокол RTSP:

GET rtsp://логин:пароль@IP-адрес:554/archive/hosts/SERVER1/DeviceIpint.4/SourceEndpoint.video:0:0/20170112T113526

GET rtsp://логин:пароль@IP-адрес:80/rtspproxy/archive/hosts/SERVER1/DeviceIpint.4/SourceEndpoint.video:0:0/20170115T113526

### 2.1.9.7 Получение информации об архивном потоке

GET http://IP-адрес:порт/префикс/archive/media/rendered-info/UUID - получение информации о последнем отображенном кадре.

Здесь UUID - уникальный идентификатор архивного потока для которого запрашивается информация.

Доступна следующая информация о кадре:

**timestamp** - временная метка кадра.

**Пример запроса:**

GET http://IP-адрес:порт/префикс/archive/media/rendered-info/22996cea31-91c4-9a46-9269-48b998fd2f29

**Пример ответа:**

```
{
  "timestamp": "20110408T103627.048"
}
```

### 2.1.9.8 Управление архивным потоком

GET http://IP-адрес:порт/префикс/archive/media/stop/UUID - остановка архивного потока с идентификатором UUID.

После успешного выполнения возвращается информация о последнем кадре.

**Примечание**  
Команда остановки архивного потока не применима для видео в формате rtsp.  
Для видео в формате hls команда stop разрывает соединение со службой NGP.

### 2.1.9.9 Покадровый просмотр архива

#### 2.1.9.9.1 Получение кадра по моменту времени

GET http://IP-адрес:порт/префикс/archive/media/VIDEOSOURCEID/STARTTIME - получение кадра, соответствующего моменту времени STARTTIME. Кадр возвращается в формате JPEG.

#### 2.1.9.9.2 Получение времени регистрации кадра

GET http://IP-адрес:порт/префикс/archive/contents/frames/VIDEOSOURCEID/ENDTIME/BEGINTIME?limit=COUNT - получение времени регистрации кадров, находящихся в архиве. Семантика параметров описана в разделе [Покадровый просмотр архива](#). Значение по умолчанию для параметра limit равно 250. Сервер не обязан следовать заданному клиентом значению limit и может по своему усмотрению вернуть меньшее количество результатов.

В возвращаемом ответе (json объекте) массив, содержащий моменты времени, соответствующие кадрам, помещается в свойство с именем **frames**.

В возвращаемом ответе (json объекте) отдельное свойство **more** - булевское значение, сигнализирующее о том, были ли выбран указанный в запросе отрезок времени полностью (false), или остались кадры, timestamp-ы которых не были возвращены из-за достижения предельного числа возвращаемых значений.

#### Пример запроса:

GET http://IP-адрес:порт/префикс/archive/contents/frames/SERVER1/DeviceIpint.2/SourceEndpoint.video:0:0/20101230T103943.000/20101230T103952.000?limit=3

#### Пример ответа:

```
{
  "frames" :
  [ "20101230T103951.800", "20101230T103951.760", "20101230T103951.720" ],
  "more" : false
}
```

### 2.1.10 Получение подписанных ссылок на видео потоки

Для получения заранее авторизованных и подписанных ссылок на видео потоки (как живого видео, так и архивного) необходимо в запрос добавить 2 параметра:

- **enable\_token\_auth** - включить авторизацию по токenu =1.
- **valid\_token\_hours** - время действительности подписи в часах. Максимальное значение - неделя. Значение по-умолчанию 12 часов.

#### Пример:

http://127.0.0.1:8000/live/media/NGP/DeviceIpint.60/SourceEndpoint.video:0:0?w=800&h=0&format=mjpeg&vc=3&enable\_token\_auth=1&valid\_token\_hours=1

- ✓ Получение живого потока от видеокamеры
- Получение архивного потока

### 2.1.11 Получение комментариев из архива

- ⚠ Axon Next 4.1.1.7766 и старше

GET http://IP-адрес:порт/префикс/archive/contents/bookmarks/ENDTIME/BEGINTIME?offset=3&limit=3

где

- **ENDTIME** и **BEGINTIME** задает интервал проверки архива (см. [Получение содержания архивов](#)).
- **offset** - смещение, количество комментариев.
- **limit** - предел полученных комментариев.

Пример запроса:

GET http://127.0.0.1:80/archive/contents/bookmarks/past/future?offset=3&limit=3

Пример ответа:

```
{
  "events" : [
    {
      "begins_at" : "20181011T154207.636",
      "comment" : "test",
      "endpoint" : "hosts/DESKTOP-LCI1BB0/DeviceIpint.1/SourceEndpoint.video:0:0",
      "ends_at" : "20181011T154207.636",
      "is_protected" : false,
      "storage_id" : "hosts/DESKTOP-LCI1BB0/MultimediaStorage.DodgerBlue/MultimediaStorage",
      "timestamp" : "20181011T160346.521000"
    },
    {
      "begins_at" : "20181011T155003.530",
      "comment" : "test",
      "endpoint" : "hosts/DESKTOP-LCI1BB0/DeviceIpint.1/SourceEndpoint.video:0:0",
      "ends_at" : "20181011T155003.530",
      "is_protected" : false,
      "storage_id" : "hosts/DESKTOP-LCI1BB0/MultimediaStorage.DodgerBlue/MultimediaStorage",
      "timestamp" : "20181011T160350.860000"
    },
    {
      "begins_at" : "20181016T085753.537",
      "comment" : "protected footage",
      "endpoint" : "hosts/DESKTOP-LCI1BB0/DeviceIpint.1/SourceEndpoint.video:0:0",
      "ends_at" : "20181016T085753.537",
      "is_protected" : true,
      "storage_id" : "hosts/DESKTOP-LCI1BB0/MultimediaStorage.DodgerBlue/MultimediaStorage",
      "timestamp" : "20181016T085834.525000"
    }
  ]
}
```

где,

- **begins\_at** и **ends\_at** - время начала и окончания комментария.
- **endpoint** - источник.
- **comment** - текст комментария.
- **is\_protected** - защита от стирания.

- **storage\_id** - архив.
- **timestamp** - дата добавления комментария.

## 2.1.12 Поиск в архиве

### 2.1.12.1 Общий интерфейс

#### 2.1.12.1.1 Запрос на поиск

##### 2.1.12.1.1.1 Поиск по одному источнику

Метод POST `http://IP-адрес:порт/префикс/search/(auto|face|vmda|stranger)/DETECTORID/BEGINTIME/ENDTIME`

где

- **auto|face|vmda|stranger** - тип поиска.
- **DETECTORID** - трехкомпонентный идентификатор endpoint-детектора (HOSTNAME/AVDetector.ID/EventSupplier для случаев auto и face, HOSTNAME/AVDetector.ID/SourceEndpoint.vmda для случая vmda, см. [Получение списка детекторов видеокамеры](#)).
- **ENDTIME, BEGINTIME** - время в формате ISO.

Для типов поиска auto и face также поддерживается запрос для поиска в рамках компьютера, имеющий следующую структуру:

`http://localhost/префикс/search/(auto|face)/HOSTID/BEGINTIME/ENDTIME,`

где **HOSTID** - имя компьютера.

##### 2.1.12.1.1.2 Поиск по нескольким источникам

Метод POST `http://IP-адрес:порт/префикс/search/(auto|face|vmda|stranger)/BEGINTIME/ENDTIME`

Этот тип запроса всегда принимает JSON в теле POST, который должен содержать как минимум одну секцию следующего вида:

```
"sources": [
    "hosts/Server1/AVDetector.1/EventSupplier"
]
```

При выполнении запроса на поиск JSON должен содержать изображение в формате base64.

```
{
    "sources": [
        "hosts/Server1/AVDetector.1/EventSupplier",
        "hosts/Server1/AVDetector.2/EventSupplier"
    ],
    "image" : "base64 encoded image"
}
```

##### 2.1.12.1.1.3 Результат

Результатом запросов будет либо возврат ошибки, либо ответ следующего вида:

```
HTTP/1.1 202 Accepted
Connection: Close
Location: /search/(auto|face|vmda|stranger)/GUID
Cache-Control: no-cache
```

Поле **Location** содержит идентификатор для последующего доступа к результатам поиска. Например, /search/vmda/3dc15b75-6463-4eb1-ab2d-0eb0a8f54bd3

Получение кода **Accepted** не гарантирует успешное выполнение поиска. Данный код означает, что команда принята в работу.

Возможные коды ошибок:

**400** - неправильно сформированный запрос.

**500** - внутренняя ошибка Сервера.

### 2.1.12.1.2 Запрос результатов поиска

Метод GET http://IP-адрес:порт/search/(auto|face|vmda|stranger)/GUID/result?offset=0&limit=10

где **/search/(auto|face|vmda)/GUID** - результат выполнения POST команды (см. [Запрос на поиск](#)).

**limit** (значение по умолчанию uint32\_t::max()) - максимальное количество событий, возвращаемых запросом.

**offset** (значение по умолчанию 0) - смещение в результирующей выборке.

Вид возвращаемых результатов зависит от типа поиска.

Запрос может вернуть два успешных статуса:

**206** - поиск ещё не завершён. Необходимо повторять запросы результатов до тех пор, пока не будет возвращён код 200. Для экономии вычислительных ресурсов рекомендуется устанавливать задержку между повторными запросами..

**200** - поиск окончен.

Возможные коды ошибок:

**400** - неправильно сформированный запрос.

**404** - заданный **offset** превышает текущее количество результатов или запрошенный идентификатор поиска (**GUID**) не найден.

Пример:

GET http://192.168.25.112:8001/search/face/49ded146-3912-4a2f-8e70-6ecfbcacdea/result

### 2.1.12.1.3 Завершение поиска

Метод DELETE http://IP-адрес:порт/search/(auto|face|vmda|stranger)/GUID

где **/search/(auto|face|vmda)/GUID** - результат выполнения POST команды (см. [Запрос на поиск](#)).


Данная команда завершает операцию поиска и освобождает используемые ресурсы. После ее выполнения результаты поиска доступны не будут.

Возможные коды ошибок:

**400** - неправильно сформированный запрос.

### 2.1.12.2 API Поиск по лицам

Тело POST запроса (см. [Запрос на поиск](#)), начинающего поиск, должно содержать бинарные данные, которые содержат изображение искомого лица в формате jpeg.

 **Внимание!**  
Если оставить тело POST запроса пустым, то поиск вернет все сработки по распознанным лицам.

В строке поиска дополнительно задается параметр **accuracy** - точность распознавания в диапазоне [0,1] (**1** - полное совпадение). Если данный параметр не задавать, будет использовано значение по умолчанию - **0.9**

Результатом поиска является JSON следующего вида:

```

{
  "events" : [
    {
      "accuracy" : 0.90591877698898315,
      "origin" : "hosts/SERVER1/DeviceIpint.2/SourceEndpoint.video:0:0",
      "position" : {
        "bottom" : 0.10694444444444445,
        "left" : 0.69687500000000002,
        "right" : 0.74687500000000007,
        "top" : 0.01805555555555554
      },
      "timestamp" : "20160914T085307.499000"
    },
    {
      "accuracy" : 0.90591877698898315,
      "origin" : "hosts/SERVER1/DeviceIpint.2/SourceEndpoint.video:0:0",
      "position" : {
        "bottom" : 0.10694444444444445,
        "left" : 0.69687500000000002,
        "right" : 0.74687500000000007,
        "top" : 0.01805555555555554
      },
      "timestamp" : "20160914T085830.392000"
    }
  ]
}

```

где

- **origin** - канал камеры с которого поступает видео поток для анализа.
- **timestamp** - время кадра на котором детектор обнаружил лицо.
- **accuracy** - точность распознавания в диапазоне [0,1], 1 - полное совпадение.
- **position** - координаты рамки, определяющей положение лица на кадре.

### 2.1.12.3 API Поиск лиц "свой"- "чужой"

Данный тип поиска сравнивает каждое распознанное лицо со всеми лицами в базе данных одной видеокамеры за 30 дней (или за текущую глубину архива, если она меньше 30 дней) и устанавливает количество дней, в которых искомое лицо было зафиксировано видеокамерой. По заданным критериям поиск определяет искомое лицо "своим" или "чужим".

Для запуска поиска используется POST запрос (см. [Запрос на поиск](#)), тип поиска **stranger**, тело запроса пустое.

При этом доступны следующие параметры:

- **accuracy** - позволяет задать степень похожести лица в диапазоне от **0** до **1** (**1** - полное совпадение). В случае отсутствия параметра будет использовано значение по умолчанию: **0.9**. Если сравниваемое лицо попадало в поле зрения видеокамеры в определенный день и было распознано с точностью, не менее указанной, то считается, что данное лицо в этот день присутствовало. В противном случае, алгоритм считает, что данное лицо в этот день отсутствовало.
- **threshold** - определяет пороговую величину для принятия лица "чужим". Значение задается в диапазоне от **0** до **1** и определяет необходимое число дней, в который искомое лицо отсутствовало, чтобы считаться "чужим" по следующей формуле:  $30 - 30 * \text{threshold}$ . Например, значение **0.8** можно интерпретировать как "искомый объект появлялся в области поиска в течении  $(30 - 30 * 0.8 = 6)$  дней". Все лица, которые появлялись 6 и более дней будут определяться как "свои", остальные - как "чужие",
- **op** - определяет направление операции поиска.  
Допустимые значения:  
**lt** - поиск "своих" лиц (исходя из определения параметром **threshold**).  
**gt** - поиск "чужих" лиц.

**⚠ Внимание!**

Параметры **threshold** и **op** должны использоваться в **только** связке. Если любой из параметров не задан или содержит некорректное значение, оба будут проигнорированы.

Результатом поиска является JSON следующего вида:

```
{
  "events" : [
    {
      "rate" : 0.90591877698898315,
      "origin" : "hosts/SERVER1/DeviceIpint.2/SourceEndpoint.video:0:0",
      "position" : {
        "bottom" : 0.10694444444444445,
        "left" : 0.69687500000000002,
        "right" : 0.74687500000000007,
        "top" : 0.01805555555555554
      },
      "timestamp" : "20160914T085307.499000"
    },
    {
      "rate" : 0.90591877698898315,
      "origin" : "hosts/SERVER1/DeviceIpint.2/SourceEndpoint.video:0:0",
      "position" : {
        "bottom" : 0.10694444444444445,
        "left" : 0.69687500000000002,
        "right" : 0.74687500000000007,
        "top" : 0.01805555555555554
      },
      "timestamp" : "20160914T085830.392000"
    }
  ]
}
```

где

- **origin** - канал камеры, с которого поступает видео поток для анализа.
- **timestamp** - время кадра, на котором детектор обнаружил лицо
- **rate** - коэффициент принятия лица "чужим", значение в диапазоне [0,1]. **1** - максимально "чужое" лицо.
- **position** - координаты рамки, определяющей положение лица на кадре

#### 2.1.12.4 Определение признака "свой"- "чужой" по изображению

##### ✓ API Поиск лиц "свой"- "чужой"

Тело POST запроса начинающего поиск должно содержать бинарные данные, которые содержат изображение искомого лица в формате jpeg. Сам запрос может быть представлен 2 вариантами:

1. POST http://IP-адрес:порт/префикс/faceAppearanceRate/DETECTORID/BEGINTIME/ENDTIME  
где  
**DETECTORID** - трехкомпонентный идентификатор endpoint-а детектора (HOSTNAME/AVDetector.ID/EventSupplier).  
**ENDTIME, BEGINTIME** - время в формате ISO.
2. POST http://IP-адрес:порт/префикс/faceAppearanceRate/HOSTID/BEGINTIME/ENDTIME  
где **HOSTID** - имя компьютера.

Дополнительно задается параметр **accuracy** - точность распознавания в диапазоне [0,1] (**1** - полное совпадение). Если данный параметр не задавать, будет использовано значение по умолчанию - **0.9**

Данный запрос выполняется синхронно и возвращает JSON следующего формата:

```
{
  "rate": 0.13333334028720856
}
```

где **rate** - коэффициент принятия лица "чужим", значение в диапазоне [0,1]. **1** - максимально "чужое" лицо.

### 2.1.12.5 API Поиск по номерам

Тело POST запроса (см. [Запрос на поиск](#)), начинающего поиск, должно содержать JSON следующего вида:

```
{
  "plate": "mask"
}
```

где параметр **plate** определяет маску поиска. Формат маски соответствует используемому в GUI (см. [Поиск по номерам](#)).

#### **Внимание!**

Если оставить тело POST запроса пустым, то поиск вернет все сработки по распознанным номерам.

Результатом поиска является JSON следующего вида:

```
{
  "events" : [
    {
      "origin" : "hosts/V-SHMELEV/DeviceIpint.4/SourceEndpoint.video:0:0",
      "plates" : [ "T470PM197", "T470PM19" ],
      "timestamp" : "20160921T084300.235000"
    },
    {
      "origin" : "hosts/V-SHMELEV/DeviceIpint.4/SourceEndpoint.video:0:0",
      "plates" : [ "T715EP199", "T715EP14" ],
      "timestamp" : "20160921T084301.795000"
    },
    {
      "origin" : "hosts/V-SHMELEV/DeviceIpint.4/SourceEndpoint.video:0:0",
      "plates" : [ "Y497XY197" ],
      "timestamp" : "20160921T084336.915000"
    }
  ]
}
```

где

- **origin** - канал камеры, с которого поступает видео поток для анализа.
- **timestamp** - время кадра, на котором детектор обнаружил номер.
- **plates** - список предлагаемых гипотез.

Для получения развернутого ответа в конце запроса необходимо добавить параметр **result\_type=full**.

Пример запроса:

```
http://localhost/search/auto/Server-1/AVDetector.1/EventSupplier/past/future?
result_type=full
```

Пример ответа:



```

{
  "events" : [
    {
      "Direction" : 0,
      "Hypotheses" : [
        {
          "OCRQuality" : 50,
          "PlateCountry" : "ru",
          "PlateFull" : "E733XA97",
          "PlateRectangle" : [
            0.40104166666666669,
            0.52941176470588236,
            0.45000000000000001,
            0.55147058823529416
          ],
          "TimeBest" : "20180730T094220.010000"
        },
        {
          "OCRQuality" : 32,
          "PlateCountry" : "ru",
          "PlateFull" : "*E733X*9",
          "PlateRectangle" : [
            0.40104166666666669,
            0.52941176470588236,
            0.45000000000000001,
            0.55147058823529416
          ],
          "TimeBest" : "20180730T094220.010000"
        },
        {
          "OCRQuality" : 38,
          "PlateCountry" : "ru",
          "PlateFull" : "E733XA***",
          "PlateRectangle" : [
            0.40104166666666669,
            0.52941176470588236,
            0.45000000000000001,
            0.55147058823529416
          ],
          "TimeBest" : "20180730T094220.010000"
        }
      ],
      "TimeBegin" : "20180730T094219.610000",
      "TimeEnd" : "20180730T094220.050000",
      "detector_type" : "plateRecognized",
      "origin_id" : "hosts/A-KULOV/DeviceIpint.1/SourceEndpoint.video:0:0",
      "phase" : 0,
      "timestamp" : "20180730T094220.010000",
      "ts_vector_body" : "E733XA97 EZERZER 7ONEZER 3TWOZER 3THRZER XFOUZER AFIVZER 9SIXZER
7SEVZER 8LENGTHZER *E733X*9 *ZERONE EONEONE 7TWOONE 3THRONE 3FOUONE XFIVONE *SIXONE 9SEVONE
8LENGHTHNE E733XA*** EZERTWO 7ONETWO 3TWOTWO 3THRTWO XFOUTWO AFIVTWO *SIXTWO *SEVTWO *EIGTWO
9LENGHTHWO"
    },
  ],
}

```

### 2.1.12.6 API Интеллектуальный поиск MomentQuest (VMDA)

Тело POST запроса (см. [Запрос на поиск](#)), начинающего поиск, должно содержать JSON одного из двух видов:

1. Конструктор, описывающий параметры для построения запроса к базе метаданных.  
Запрос на поиск состоит из трех логических частей:
  - a. Тип запроса (queryType, см. [Типы запросов и их параметры](#))
  - b. Параметры специфичные для заданного типа запроса (figures, queryProperties, см. [Дополнительные условия](#))
  - c. Универсальные дополнительные условия фильтрации (objectProperties, conditions, см. [Дополнительные условия](#))
2. Непосредственный запрос на языке базы метаданных.

```
{
  "query": "figure
fZone=polygon(0.4647676,0.3973333,0.7946027,0.5493333,0.8650675,0.7946666,0.4647676,0.7946
666); figure fDir=(ellipses(-10000, -10000, 10000, 10000) - ellipses(-0, -0, 0,
0));set r = group[obj=vmda_object] { res = or(fZone((obj.left + obj.right) / 2,
obj.bottom)) }; result = r.res;"
}
```

**⚠ Внимание!**

При наличии во входном JSON секций конструктора и непосредственного запроса одновременно, непосредственный запрос имеет больший приоритет.

**⚠ Внимание!**

Если оставить тело POST запроса пустым, то поиск вернет все интервалы сработок.

**ℹ Примечание**

Для поиска по данным [оффлайн-аналитики](#) следует использовать запрос вида:

```
POST /search/vmda/SERVER-NAME/OfflineAnalytics.c95ad5a581094845995ee28a7f097797/
SourceEndpoint.vmda:AVDetector:1/past/future
```

Этот запрос будет работать даже если удален архив Axhon Next, но метаданные в VMDA сохранены. Следует обратить внимание на то, что идентификатор объектов должен быть записан без префикса **hosts/**  
Правильный запрос: /search/vmda/SERVER-NAME/OfflineAnalytics...  
Ошибочный запрос: /search/vmda/hosts/SERVER-NAME/OfflineAnalytics...

Результатом поиска является JSON следующего вида:

```
{
  "intervals" : [
    {
      "endTime" : "20160919T064640.430000",
      "startTime" : "20160919T064636.390000"
    },
    {
      "endTime" : "20160919T073204.113000",
      "startTime" : "20160919T073201.513000"
    }
  ]
}
```

где **Intervals** – набор интервалов времени, в которые истинно условие поиска.

### 2.1.12.6.1 Типы запросов и их параметры

#### На странице:

- Пребывание объекта в зоне (queryType="zone")
- Перемещение объекта из одной зоны в другую (queryType="transition")
- Пересечение линии (queryType="line")

#### 2.1.12.6.1.1 Пребывание объекта в зоне (queryType="zone")

Обязательным параметром является **figures/shape**, который задает зону, в которой должен находиться объект в виде списка вершин полигона. Координаты задаются в долях от ширины/высоты кадра (значения от 0 до 1). Начало координат в левом верхнем углу кадра. Первая координата в каждой паре отсчитывается по горизонтальной оси, а вторая – по вертикальной. Это позволяет не привязываться к конкретному разрешению видеокамеры.

Самый простой запрос выглядит следующим образом:

```
{
  "queryType": "zone",
  "figures": [
    {
      "shape": [
        [0.3, 0.3],
        [0.7, 0.3],
        [0.7, 0.7],
        [0.3, 0.7]
      ]
    }
  ]
}
```

Здесь зона описывает прямоугольник в центре поля зрения камеры.

Необязательным параметром запроса является **queryProperties/action**:

- Если этот параметр не задан, ищутся объекты находящиеся в зоне.
- При queryProperties/action="enter" ищутся объекты вошедшие в зону.
- При queryProperties/action="exit" ищутся объекты покинувшие в зону.

Пример запроса объектов вошедших в зону:

```

{
  "queryType": "zone",
  "figures": [
    {
      "shape": [
        [0.3, 0.3],
        [0.7, 0.3],
        [0.7, 0.7],
        [0.3, 0.7]
      ]
    }
  ],
  "queryProperties": {
    "action": "enter"
  }
}

```

#### 2.1.12.6.1.2 Перемещение объекта из одной зоны в другую (queryType="transition")

Обязательный параметр один — **figures**. Он должен содержать две зоны описывающие начальную и конечную зоны.

Необязательных параметров нет.

Поиск объектов переместившихся из левой половины кадра в правую:

```

{
  "queryType": "transition",
  "figures": [
    {
      "shape": [
        [0, 0],
        [0.45, 0],
        [0.45, 1],
        [0, 1]
      ]
    },
    {
      "shape": [
        [0.55, 0],
        [1, 0],
        [1, 1],
        [0.55, 1]
      ]
    }
  ]
}

```

#### 2.1.12.6.1.3 Пересечение линии (queryType="Line")

Обязательный параметр **figures** определяет отрезок пересечение которого приводит к срабатыванию условия. Отрезок должен быть задан ровно двумя точками.

```

{
  "queryType": "line",
  "figures": [
    {
      "shape": [
        [0.5, 0.8],
        [0.5, 0.2]
      ]
    }
  ]
}

```

Необязательный параметр **queryProperties/direction** задает направление пересечения линии.

- Если параметр не задан, в результаты попадут объекты пересекшие линию в любом направлении.
- `queryProperties/direction="left"` означает, что объект должен пересечь линию справа налево, если смотреть из первой точки отрезка.
- `queryProperties/direction="right"` означает, что объект должен пересечь линию слева направо, если смотреть из первой точки отрезка.

```

{
  "queryType": "line",
  "figures": [
    {
      "shape": [
        [0.5, 0.8],
        [0.5, 0.2]
      ]
    }
  ],
  "queryProperties": {
    "direction": "left"
  }
}

```

#### 2.1.12.6.2 Дополнительные условия

##### На странице:

- Тип объекта (`objectProperties/category`)
- Размер объекта (`objectProperties/size`)
- Цвет объекта (`objectProperties/color`)
- Скорость (`conditions/velocity`)
- Направление движения (`conditions/directions`)
- Длительность (`conditions/duration`)
- Количество объектов (`condtions/count`)

Дополнительные условия подходят ко всем видам запросов. Условия всегда объединяются логическим «И». Например запрос «Объект высотой не более четверти кадра находящийся в центре поля зрения камеры в течении 5 секунд» выглядит так:

```
{
  "queryType": "zone",
  "figures": [
    {
      "shape": [
        [0.3, 0.3],
        [0.7, 0.3],
        [0.7, 0.7],
        [0.3, 0.7]
      ]
    }
  ],
  "objectProperties": {
    "size": {
      "height": [0, 0.25]
    }
  },
  "conditions": {
    "duration": 5
  }
}
```

#### 2.1.12.6.2.1 Тип объекта (`objectProperties/category`)

Объект может быть оставленным (`abandoned`) или движущимся (`face, human, group, vehicle`). В запросе нельзя смешивать `abandoned` с другими типами объектов (иначе требование `abandoned` будет проигнорировано).

Поиск оставленных объектов в любой точке кадра:

```
{
  "queryType": "zone",
  "figures": [
    {
      "shape": [
        [0, 0],
        [1, 0],
        [1, 1],
        [0, 1]
      ]
    }
  ],
  "objectProperties": {
    "category": ["abandoned"],
  }
}
```

Поиск одиночных или небольших групп людей, пересекших заданную линию:

```

{
  "queryType": "line",
  "figures": [
    {
      "shape": [
        [0.5, 0.8],
        [0.5, 0.2]
      ]
    }
  ]
  "objectProperties": {
    "category": ["human", "group"],
  }
}

```

#### 2.1.12.6.2.2 Размер объекта (objectProperties/size)

Задаёт минимальные и максимальные ширину и высоту объекта.

Например, чтобы найти объекты не больше четверти кадра в высоту можно использовать такой запрос:

```

{
  "queryType": "zone",
  "figures": [
    {
      "shape": [
        [0.3, 0.3],
        [0.7, 0.3],
        [0.7, 0.7],
        [0.3, 0.7]
      ]
    }
  ],
  "objectProperties": {
    "size": {
      "width": [0, 1],
      "height": [0, 0.25]
    }
  }
}

```

Так как не обязательно указывать оба габарита, этот запрос будет эквивалентен предыдущему:

```

{
  "queryType": "zone",
  "figures": [
    {
      "shape": [
        [0.3, 0.3],
        [0.7, 0.3],
        [0.7, 0.7],
        [0.3, 0.7]
      ]
    }
  ],
  "objectProperties": {
    "size": {
      "height": [0, 0.25]
    }
  }
}

```

#### 2.1.12.6.2.3 Цвет объекта (`objectProperties/color`)

Задаёт минимальные и максимальные координаты цвета объекта в пространстве HSV. hue измеряется в градусах (от 0 до 360), a saturation и brightness в долях от 0 до 1.

Запрос на получение ярко зеленых объектов в зоне:

```

{
  "queryType": "zone",
  "figures": [
    {
      "shape": [
        [0.3, 0.3],
        [0.7, 0.3],
        [0.7, 0.7],
        [0.3, 0.7]
      ]
    }
  ],
  "objectProperties": {
    "color": {
      "hue": [75, 135],
      "saturation": [0.5, 1],
      "brightness": [0.5, 1]
    }
  }
}

```

В пространстве HSV темные почти черные цвета могут обладать любыми тоном и насыщенностью. Поэтому для поиска черных объектов запрос должен выглядеть так:



```

{
  "queryType": "zone",
  "figures": [
    {
      "shape": [
        [0.3, 0.3],
        [0.7, 0.3],
        [0.7, 0.7],
        [0.3, 0.7]
      ]
    }
  ],
  "objectProperties": {
    "color": {
      "hue": [0, 360],
      "saturation": [0, 1],
      "brightness": [0, 0.2]
    }
  }
}

```

Аналогичный запрос для белых объектов:

```

{
  "queryType": "zone",
  "figures": [
    {
      "shape": [
        [0.3, 0.3],
        [0.7, 0.3],
        [0.7, 0.7],
        [0.3, 0.7]
      ]
    }
  ],
  "objectProperties": {
    "color": {
      "hue": [0, 360],
      "saturation": [0, 0.1],
      "brightness": [0.8, 1]
    }
  }
}

```

#### 2.1.12.6.2.4 Скорость (**conditions/velocity**)

Задаёт минимальную и максимальную скорость объекта.

Измеряется в долях кадра в секунду. То есть скорость объекта переместившегося от левой границы кадра к правой за одну секунду равна 1.

```

{
  "queryType": "zone",
  "figures": [
    {
      "shape": [
        [0.3, 0.3],
        [0.7, 0.3],
        [0.7, 0.7],
        [0.3, 0.7]
      ]
    }
  ],
  "conditions": {
    "velocity": [0.25, 1]
  }
}

```

#### 2.1.12.6.2.5 Направление движения (conditions/directions)

Задаёт направление движения объекта в виде массива углов. Углы измеряются в радианах и отсчитываются от оси направленной вправо по часовой стрелке.

Таким образом запрос на получение объектов двигавшихся вправо  $\pm 45^\circ$  выглядит так:

```

{
  "queryType": "zone",
  "figures": [
    {
      "shape": [
        [0.3, 0.3],
        [0.7, 0.3],
        [0.7, 0.7],
        [0.3, 0.7]
      ]
    }
  ],
  "conditions": {
    "directions": [
      [315, 45]
    ]
  }
}

```

Обратите внимание, что угол  $45^\circ$  --  $315^\circ$  захватывает все направления кроме «вправо».

Если необходимо найти объекты двигавшиеся преимущественно в горизонтальном направлении понадобится задать уже два угла:

```

{
  "queryType": "zone",
  "figures": [
    {
      "shape": [
        [0.3, 0.3],
        [0.7, 0.3],
        [0.7, 0.7],
        [0.3, 0.7]
      ]
    }
  ],
  "conditions": {
    "directions": [
      [315, 45],
      [135, 225]
    ]
  }
}

```

#### 2.1.12.6.2.6 Длительность (conditions/duration)

Задаёт время в секундах в течении которых объект должен удовлетворять всем поставленным условиям непрерывно.

С помощью этого условия может быть выражен запрос «длительное пребывание в зоне»:

```

{
  "queryType": "zone",
  "figures": [
    {
      "shape": [
        [0.3, 0.3],
        [0.7, 0.3],
        [0.7, 0.7],
        [0.3, 0.7]
      ]
    }
  ],
  "conditions": {
    "duration": 5
  }
}

```

#### 2.1.12.6.2.7 Количество объектов (conditions/count)

Задаёт минимальное необходимое количество объектов одновременно удовлетворяющих остальным условиям запроса.

Обычно применяется для поиска большого количества объектов в зоне, например:

```
{
  "queryType": "zone",
  "figures": [
    {
      "shape": [
        [0.3, 0.3],
        [0.7, 0.3],
        [0.7, 0.7],
        [0.3, 0.7]
      ]
    }
  ],
  "conditions": {
    "count": 3
  }
}
```

## 2.1.13 Получение списка групп и их содержимого

### 2.1.13.1 Получение списка всех доступных групп

GET <http://IP-адрес:порт/префикс/group>

Пример ответа:

```

{
  "groups" : [
    {
      "Brief" : "Group1",
      "Description" : "",
      "Id" : "35fc84a0-2280-4b30-acd2-cc8419a2dc68",
      "ObjectCount" : "14"
      "groups" : [
        {
          "Brief" : "Group2",
          "Description" : "",
          "Id" : "dac24803-313c-43ab-aa9a-276922a55cb6",
          "ObjectCount" : "5"
          "groups" : []
        },
        {
          "Brief" : "Group3",
          "Description" : "",
          "Id" : "13764152-6910-44b6-99b5-f74641ad4a14",
          "ObjectCount" : "3"
          "groups" : [
            {
              "Brief" : "Group4",
              "Description" : "Group4",
              "Id" : "9a64e2a0-eb92-4adc-bc4f-81d30ceb6c2f",
              "ObjectCount" : "6"
              "groups" : []
            }
          ]
        }
      ]
    }
  ]
}

```

**ObjectCount** - количество видеокамер в данной группе.

### 2.1.13.2 Получение содержимого группы

GET http://IP-адрес:порт/префикс/group/GROUPID

где **GROUPID** - значение поля **Id**, полученного с помощью предыдущего вызова.

Пример ответа:

```

{
  "members" : [ "hosts/SERVER1/DeviceIpint.1/SourceEndpoint.video:0:0" ]
}

```

### 2.1.13.3 Получение списка групп, содержащих указанную видеокамеру

GET http://IP-адрес:порт/префикс/group/contains/VIDEOSOURCEID

где **VIDEOSOURCEID** - трехкомпонентный идентификатор endpoint-а источника (см. [Получение списка источников видео \(камер\)](#)).

Пример:

```
http://localhost:8000/group/contains/SERVER1/DeviceIpint.1/SourceEndpoint.video:0:0
```

Пример ответа:

```
{
  "groups" : [
    "35fc84a0-2280-4b30-acd2-cc8419a2dc68",
    "13764152-6910-44b6-99b5-f74641ad4a14",
    "dac24803-313c-43ab-aa9a-276922a55cb6"
  ]
}
```

## 2.1.14 Экспорт

### На странице:

- [Запуск экспорта](#)
- [Запрос статуса экспорта](#)
- [Завершение экспорта](#)
- [Скачивание файла](#)

### 2.1.14.1 Запуск экспорта

Экспорт инициируется с помощью одного из следующих POST запросов:

http://IP-адрес:порт/префикс/export/archive/SERVER1/VIDEOSOURCEID/BEGINTIME/ENDTIME - экспорт из архива  
 http://IP-адрес:порт/префикс/export/live/SERVER1/VIDEOSOURCEID/BEGINTIME/ENDTIME - экспорт живого потока

где **BEGINTIME** и **ENDTIME** задают время в формате **YYYYMMDDTHHMMSS**. Если **BEGINTIME** больше **ENDTIME**, то значения поменяются местами. Для экспорта кадра **BEGINTIME** должно быть равно **ENDTIME**.

Для описания рамок и масок используются сложные структуры данных, которые можно разделить на следующие типы:

- **point** - задается с помощью следующего синтаксиса *x,y*: пример - [0.5, 0.5].
- **area** - задает квадратную рамку, определяется двумя points с разделителем **!**. Пример - [[0.5, 0.5], [0.4,0.4]].
- **polygon** - задает замкнутую кривую, содержит как минимум 3 точки заключенные в []. Пример - [[0.5, 0.5], [0.4,0.4],[0.3,0.3]].
- **mask** - коллекция polygons. Пример - [[[0.5, 0.5],[0.6, 0.6],[0.7, 0.7]], [[0.1, 0.1],[0.2, 0.2],[0.3, 0.3]]].

Поддерживаемые параметры передаваемые в теле начального POST запроса:

#### Раскрыть список

1. **format** (строка) - **обязательный параметр**, доступные значения *mkv, avi, exe, jpg, pdf*. Задает формат выходного контейнера экспорта.
2. **maxfilesize** (число) - максимальный размер файла экспорта в байтах. При достижении этого размера будет создаваться новый файл. Результатом экспорта будет коллекция файлов. Значение по умолчанию - 0 (на выходе единственный файл).
3. **vc, ac** (число) - уровень качества сжатия для видео и аудио соответственно. Допустимые значения от 0 до 6 (6 - наихудшее). Тонкий нюанс - при указании 0 в значении параметра будет оставлен исходный уровень качества (скорее всего недоработка исходного API). Значение по умолчанию - 0.
4. **freq** (число) - частота кадров выходного потока. Значение по умолчанию - 0. Допустимые значения:

- a. **0** - оригинальная
  - b. **1** - половина оригинальной
  - c. **2** - четверть
  - d. **3** - осьмушка
5. **tsformat** (строка) - шаблон формата временной метки. Можно генерировать любую строку на основании [http://www.boost.org/doc/libs/1\\_55\\_0/doc/html/date\\_time/date\\_time\\_io.html](http://www.boost.org/doc/libs/1_55_0/doc/html/date_time/date_time_io.html). Значение по умолчанию: %Y-%b-%d %H:%M:%S.

**Внимание!**

Сервер не проверяет формат введенной строки.

- 6. **croparea** (area) - область кадра для экспорта (Значение по умолчанию 0,0!0,0 - весь кадр).
- 7. **maskspace** (mask) - область маскировки кадра. По умолчанию кадр не маскируется.
- 8. **color** (строка) - цвет текста для комментария и временной метки. Задается в Web-формате - #FFFFFF.
- 9. **comment** (строка) - комментарий

Параметры, имеющие смысл только для формата PDF.

- 1. **snapshotplace** (area) - положение кадра на странице
- 2. **commentplace** (area) - положение комментария на странице
- 3. **tsplace** (area) - положение временной метки на странице
- 4. **layout** (число) - ориентация страницы. Допустимые значения - **0** (портретная), **1** (альбомная).

Результатом запроса будет либо возврат ошибки, либо ответ следующего вида:

```
HTTP/1.1 202 Accepted
Connection: Close
Location: /export/3dc15b75-6463-4eb1-ab2d-0eb0a8f54bd3
Cache-Control: no-cache
```

**Примечание**

Возможные коды ошибок:

- **400** - неправильно сформированный запрос.
- **500** - внутренняя ошибка Сервера.

### 2.1.14.2 Запрос статуса экспорта

GET <http://IP-адрес:порт/префикс/export/id/status>

где **id** - значение из поля **Location** (в данном случае 3dc15b75-6463-4eb1-ab2d-0eb0a8f54bd3)

Пример ответа:

```
{
  "id": "73c2e1d2-0f8f-414c-9cc0-ac5fb43cd8dd"
  , "state": 1
  , "progress": 0.51062298
  , "error": ""
  , "files":
  [
  ]
}
```

где

- **state** определяет текущее состояние экспорта. Возможные значения:
  - 1 - экспорт выполняется
  - 2 - экспорт завершен
  - 3 - ошибка экспорта
  - 4 - недостаточно места для выполнения операции
- **progress** - прогресс сессии экспорта в диапазоне от 0 до 1.
- **error** - описание ошибки, если таковая присутствует
- **files** - список файлов, созданных в результате экспорта

### 2.1.14.3 Завершение экспорта

DELETE http://IP-адрес:порт/префикс/export/id

где **id** - значение из поля **Location**.

### 2.1.14.4 Скачивание файла

GET http://IP-адрес:порт/префикс/export/id/file?name=SERVER\_DeviceIpint.1[20160527T132900-20160527T133000].mkv

где

- **id** - значение из поля **Location**
- **file?name** - имя файла из поля **files**

### 2.1.15 Получение списка детекторов видеокамеры

GET http://IP-адрес:порт/префикс/detectors/SERVER1/DeviceIpint.N

где **N** - идентификатор видеокамеры (см. [Получение списка источников видео \(камер\)](#)).

Пример ответа:

```
{
  "detectors": [
    {
      "name": "hosts/SERVER1/AVDetector.1/EventSupplier",
      "type": "SceneDescription"
    },
    {
      "name": "hosts/SERVER1/AVDetector.12/EventSupplier",
      "type": "NullAudioDetection"
    }
  ]
}
```

Список возможных значений параметра **type**:

SceneDescription	Детектор анализа ситуации
FireDetector	Детектор дыма
SmokeDetector	Детектор огня
LprDetector	Детектор распознавания номеров
TvaFaceDetector	Детектор лиц



QualityDegradation_v2	Детектор зашумления изображения
QualityDegradation	Деградация качества
SceneChange	Изменение положения
BlurredDegradation	Детектор размытого изображения
MotionDetection	Детектор движения
CompressedDegradation	Детектор артефактов сжатия
SignalAudioDetection	Отсутствие сигнала
NoiseAudioDetection	Шум
NullAudioDetection	Отсутствие сигнала

- ✓ Детекторы анализа ситуации
- Типы видеодетекторов
- Типы аудиодетекторов

## 2.1.16 Получение информации о сработках детекторов и тревогах

### 2.1.16.1 Получение списка тревог

GET http://IP-адрес:порт/префикс/archive/events/alerts/VIDEOSOURCEID/ENDTIME/BEGINTIME?limit=COUNT&offset=COUNT - получение списка тревог. В случае, если значение limit не указано, оно считается равным 100. Поле **raisedAt** не уникальное, поэтому может понадобиться пропуск уже полученных ранее тревог от начала интервала поиска.

#### Пример ответа:

```
{
  "events":
  [
    {
      "type": "alert",
      "id": "42D43A79-90D6-4ba7-BD23-1714996A2F88",
      "raisedAt": "20101230T103950.000",
      "zone": "SERVER1/DeviceIpint.3/SourceEndpoint.video:0:0",
      "reasons": ["ruleAlert", "videoDetector"],
      "initiator": "plateRecognized",

      "reaction":
      {
        "user": "root",
        "reactedAt": "20101230T103958.000",
        "severity": "alarm"
      },
      "detectorName": "5.Распознавание номеров"
    },
    ...
  ]
}
```

```

],
"more": true
}

```

Возможные значения в массиве **reasons**: armed, disarmed, userAlert, ruleAlert, videoDetector, audioDetector, ray.

Возможные значения поля **severity**: unclassified, false, notice, warning, alarm.

Возможные значения поля **initiator**:

1. Имя пользователя, инициировавшего тревогу.
2. **Id** макрокоманды (см. [Работа с макрокомандами](#)), если тревога инициирована макрокомандой, которая была запущена НЕ детектором. При этом следует иметь в виду, что данная макрокоманда может быть уже удалена из конфигурации.
3. Тип детектора, который вызвал сработку макрокоманды, которая инициировала тревогу. Возможные типы детекторов:

группа детекторов первого уровня:

- SceneChangeDetected
- CameraBlindDetected
- MotionDetected (не маска движения, а именно сработка)
- NullAudio
- NoiseAudio
- SignalAudio
- Ray
- plateRecognized
- faceAppeared
- SmokeDetected
- FireDetected
- ImageDegradation
- BlurredDegradation

группа детекторов второго уровня

- oneLine
- comelnZone
- lostObject
- outOfZone
- longlnZone
- movelnZone
- stoplnZone

4. Иная информация:

- a. какой-либо детектор, значение типа сработки которого не стандартизировано;
- b. пользователь, который был удалён из системы.

#### **Внимание!**

Не гарантируется регистрозависимость типов сработок детекторов. Например, значения 'ComelnZone', 'comelnZone', 'COMEINZOne' следует рассматривать как одинаковые.

### 2.1.16.2 Получение списка срабатываний детекторов

GET http://IP-адрес:порт/префикс/archive/events/detectors/VIDEOSOURCEID/ENDTIME/BEGINTIME?

limit=COUNT&offset=COUNT - получение списка срабатываний детекторов. В случае, если значение limit не указано, оно считается равным 100. Поле **timestamp** не уникальное, поэтому, может понадобиться пропуск уже полученных ранее срабатываний детекторов от начала интервала поиска.

**Пример ответа:**

```

{
"events":
[
{
"id": "433d45ec-0b7f-aa43-8491-c8acb7d0ac56"
,"source": "hosts/SERVER1/Devicelpint.3/SourceEndpoint.video:0:0"

```

```

,"origin": "hosts/SERVER1/DeviceIpint.3/EventSupplier.analytics:0:motion_detection_snb_5001"
,"type": "oneLine"
,"alertState": "ended"

,"multiPhaseSyncId" : "f5294404-7cf0-419e-b133-694edb08d69e"
,"timestamp": "20120314T121512.597"
,"rectangles":
[
{
"index": "1"
,"left": "0.622086710929871"
,"top": "0.68798337459564196"
,"right": "0.65736908435821495"
,"bottom": "0.79889315128326399"
}
]
},
...
],
"more": true
}

```

**Примечание**  
 Параметр **multiPhaseSyncId** будет в ответе только для двухфазных тревог с параметром **alertState**.

В данном запросе VIDEOSOURCEID может быть:

- как обычно, трехкомпонентным, например - "HOST/DeviceIpint.2/SourceEndpoint.video:0:0";
- именем хоста, для получения событий с него;
- пустым, т.е. отсутствовать, для получения всех событий домена.

#### Примеры запросов:

http://IP-адрес:порт/префикс/archive/events/detectors/HOST/DeviceIpint.2/SourceEndpoint.video:0:0/past/future?limit=10&offset=0 - получить сработки по указанной видеокамере, отсортированные по убыванию. Максимальное количество - 10.

http://IP-адрес:порт/префикс/archive/events/detectors/**HOST**/past/future?limit=5&offset=0 - получить сработки всех детекторов, созданных на машине HOST. Максимальное количество по каждому детектору - 5.

http://IP-адрес:порт/префикс/archive/events/detectors/**HOST**/past/future?limit=5&offset=0&**type=Ray** - получить сработки всех лучей, созданных на машине HOST. Максимальное количество по каждому лучу - 5.

http://IP-адрес:порт/префикс/archive/events/detectors/past/future?limit=1&**type=Ray** Получить текущее состояние всех лучей домен.

#### Возможные значения параметра type:

- SceneChangeDetected;
- CameraBlindDetected;
- Disconnected;
- MotionDetected;
- NullAudio;
- NoiseAudio;
- SignalAudio;
- Ray;

- oneLine;
- comelnZone;
- lostObject;
- outOfZone;
- longlnZone;
- movelnZone;
- stoplnZone;
- faceAppeared;
- plateRecognized;
- BlurredDegradation;
- FireDetected;
- SmokeDetected.

**Примечание**

Если было получено значение, не входящее в указанный список, значит это сработка встроенного детектора

## 2.1.17 Управление телеметрией

### 2.1.17.1 Получение списка устройств телеметрии для источника видео

GET http://IP-адрес:порт/префикс/control/telemetry/list/OBJECTID - Получение списка устройств телеметрии для источника видео, где OBJECTID – HOSTNAME/DeviceIpint.Id.

**Пример ответа:**

```
[
"SERVER1/DeviceIpint.2/TelemetryControl.0"
]
```

В дальнейшем для обозначения устройств телеметрии вида HOSTNAME/DeviceIpint.Id/TelemetryContol.n будет использоваться шаблон TELEMETRYCONTROLID.

### 2.1.17.2 Захват сессии управления телеметрией

GET http://IP-адрес:порт/префикс/control/telemetry/session/acquire/[server\_name]/[device\_name]/[telemetry\_name]? session\_priority=[priority],

где server\_name - имя Сервера (см. [Получение списка серверов](#));

device\_name - имя видеокamеры (см. [Получение списка источников видео \(камер\)](#));

telemetry\_name - имя устройства телеметрии (см. [Получение списка устройств телеметрии для источника видео](#));

priority - приоритет управления телеметрией от 1 (максимальный) до 5 (минимальный).

Если в данный момент устройство телеметрии свободно или им управляет другой пользователь с меньшим приоритетом, то происходит захват управления и от Сервера приходит ответ:

```
{
"session_id" : [id]
}
```

где id - идентификатор сессии.

### 2.1.17.3 Поддержание актуальности сессии

GET http://IP-адрес:порт/префикс/control/telemetry/session/keepalive/[server\_name]/[device\_name]/[telemetry\_name]?session\_id=[id],

где server\_name - имя Сервера (см. [Получение списка серверов](#));

device\_name - имя видеокамеры (см. [Получение списка источников видео \(камер\)](#));

telemetry\_name - имя устройства телеметрии (см. [Получение списка устройств телеметрии для источника видео](#));

id - идентификатор захваченной сессии (см. [Захват сессии управления телеметрией](#)).

Данный запрос следует отправлять не реже чем раз в 10 секунд для поддержания актуальности сессии. В противном случае управление телеметрией будет потеряно.

### 2.1.17.4 Освобождение сессии

GET http://IP-адрес:порт/префикс/control/telemetry/session/release/[server\_name]/[device\_name]/[telemetry\_name]?session\_id=[id],

где server\_name - имя Сервера (см. [Получение списка серверов](#));

device\_name - имя видеокамеры (см. [Получение списка источников видео \(камер\)](#));

telemetry\_name - имя устройства телеметрии (см. [Получение списка устройств телеметрии для источника видео](#));

id - идентификатор захваченной сессии (см. [Захват сессии управления телеметрией](#)).

### 2.1.17.5 Управление степенями свободы

#### На странице:

- [Получение информации о степенях свободы](#)
- [Изменение наклона и поворота](#)
- [Изменение одной из степеней свободы](#)
- [Переход по точке на экране](#)
- [Увеличение области изображения](#)
- [Автоматическая фокусировка и диафрагма](#)

#### 2.1.17.5.1 Получение информации о степенях свободы

GET http://IP-адрес:порт/префикс/control/telemetry/info/TELEMETRYCONTROLID - Получение информации о степенях свободы, управление которыми поддерживается, способах управления ими (непрерывный, относительный, дискретный), предельно допустимые значения.

#### Пример ответа:

```
{
"degrees":
{
"tilt":
{
"relative": {"min": "-45", "max": "45"},
"continuous": {"min": "-10", "max": "10"}
}
}
}
```

```

},
"pan":
{
"absolute": {"min": "-170", "max": "170"},
"continuous": {"min": "-10", "max": "10"}
},
"zoom":
{
"absolute": {"min": "0", "max": "20"}
}
},
"feature": ["autoFocus", "areaZoom", "pointMove"]
}

```

**degrees** - информация о степенях свободы (tilt, pan, zoom, focus, iris). Каждая степень свободы содержит список поддерживаемых способов управления (absolute, relative, continuous).

**feature** - список поддерживаемых функций (autoFocus, autoIris, areaZoom, pointMove).

#### 2.1.17.5.2 Изменение наклона и поворота

GET http://IP-адрес:порт/префикс/control/telemetry/move/TELEMETRYCONTROLID?параметры&session\_id=N - изменение наклона, поворота (tilt, pan).

Параметры:

**mode** - способ управления (absolute, relative, continuous);

**pan, tilt** - значение для соответствующих степеней;

здесь и далее **session\_id=N** - идентификатор сессии (см. [Захват сессии управления телеметрией](#)).

**Пример запроса:** GET http://IP-адрес:порт/префикс/control/telemetry/move/HOSTNAME/DeviceIpint.25/TelemetryControl.0?mode=absolute&pan=-99&tilt=10&session\_id=0

#### 2.1.17.5.3 Изменение одной из степеней свободы

GET http://IP-адрес:порт/префикс/control/telemetry/{**степень**}/TELEMETRYCONTROLID?параметры&session\_id=N - изменение одной из степеней (zoom, focus, iris).

Параметры:

{**степень**} - изменяемая степень свободы (zoom, focus, iris);

**mode** - способ управления (absolute, relative, continuous);

**value** - значение.

**Пример запроса:**

GET http://IP-адрес:порт/префикс/control/telemetry/zoom/HOSTNAME/DeviceIpint.25/TelemetryControl.0?mode=absolute&value=6&session\_id=0 - меняется zoom;

GET http://IP-адрес:порт/префикс/control/telemetry/focus/HOSTNAME/DeviceIpint.25/TelemetryControl.0?mode=relative&value=3&session\_id=0 - меняется focus;

GET http://IP-адрес:порт/префикс/control/telemetry/iris/HOSTNAME/DeviceIpint.25/TelemetryControl.0?mode=continuous&value=1&session\_id=0 - меняется iris.

#### 2.1.17.5.4 Переход по точке на экране

GET http://IP-адрес:порт/префикс/control/telemetry/move/point/TELEMETRYCONTROLID?параметры&session\_id=N - переход по точке на экране.

Параметры:

**x,y** - значения координат точки по горизонтали и вертикали, указываемые относительно размера изображения;

**Пример запроса:**

GET http://IP-адрес:порт/префикс/control/telemetry/move/point/HOSTNAME/DeviceIpint.23/TelemetryControl.0?x=0.14&y=0.32&session\_id=0

**2.1.17.5.5 Увеличение области изображения**

GET http://IP-адрес:порт/префикс/control/telemetry/zoom/area/TELEMETRYCONTROLID?параметры&session\_id=N – увеличение (zoom) выбранной области изображения.

Параметры:

**x,y** - левый верхний угол области;

**w,h** - ширина и высота области.

Координаты и размеры указываются относительно размера изображения.

**Пример запроса:**

GET http://IP-адрес:порт/префикс/control/telemetry/zoom/area/HOSTNAME/DeviceIpint.24/TelemetryControl.0?x=0.23&y=0.089&w=0.25&h=0.25&session\_id=0

**2.1.17.5.6 Автоматическая фокусировка и диафрагма**

GET http://IP-адрес:порт/префикс/control/telemetry/auto/TELEMETRYCONTROLID?параметры&session\_id=N - авто фокусировка/диафрагма.

Параметры:

**degree** – принимает значения focus или iris.

**Пример запроса:**

GET http://IP-адрес:порт/префикс/control/telemetry/auto/HOSTNAME/DeviceIpint.24/TelemetryControl.0?degree=iris&session\_id=0

**2.1.17.6 Управление предустановками (preset)****На странице:**

- [Получение списка предустановками](#)
- [Создание и изменение предустановки](#)
- [Переход к предустановке и удаление](#)

**2.1.17.6.1 Получение списка предустановками**

GET http://IP-адрес:порт/префикс/control/telemetry/preset/info/TELEMETRYCONTROLID- получить список существующих предустановок.

**Пример запроса:**

GET http://IP-адрес:порт/префикс/control/telemetry/preset/info/HOSTNAME/DeviceIpint.23/TelemetryControl.0

**Пример ответа:**

```
{
  "0": "Коридор",
  "1": "Входная дверь",
```

```
"4": "Дырка в заборе"
}
```

### 2.1.17.6.2 Создание и изменение предустановки

GET http://IP-адрес:порт/префикс/control/telemetry/preset/set/TELEMETRYCONTROLID?параметры&session\_id=N - создать/изменить предустановку.

Параметры:

**pos** - индекс;

**label** - наименование предустановки;

здесь и далее **session\_id=N** - идентификатор сессии (см. [Захват сессии управления телеметрией](#)).

Если предустановка с указанным индексом уже существует, то его метка будет изменена.

**Пример запроса:**

```
GET http://IP-адрес:порт/префикс/control/telemetry/preset/set/HOSTNAME/DeviceIpint.23/TelemetryControl.0?
pos=0&label=Exit&session_id=0
```

### 2.1.17.6.3 Переход к предустановке и удаление

GET http://IP-адрес:порт/префикс/control/telemetry/preset/{действие}/TELEMETRYCONTROLID?параметры&session\_id=N - перейти или удалить предустановку.

Параметры:

**{действие}** – принимает значение **go** или **remove**, используется чтобы перейти к/удалить предустановку;

**pos** - индекс предустановки.

**Пример запроса:**

Осуществляется переход к уже созданной предустановке с индексом 1:

```
GET http://IP-адрес:порт/префикс/control/telemetry/preset/go/HOSTNAME/DeviceIpint.23/TelemetryControl.0?
pos=1&session_id=0
```

Удаление предустановки с индексом 2:

```
GET http://IP-адрес:порт/префикс/control/telemetry/preset/remove/HOSTNAME/DeviceIpint.23/TelemetryControl.0?
pos=2&session_id=0
```

### 2.1.17.7 Получение информации об ошибках

При возникновении ошибок при запросах к телеметрии будет получен ответ вида { "error\_code" : [числовой код ошибки] }

Возможные коды ошибок:

- 1** - Общая ошибка, подробности в логах Сервера.
- 2** - Неверно заданы параметры запроса.
- 3** - Сессия управления телеметрией недоступна.
- 4** - Ошибка при управлении предустановками.

### 2.1.18 Работа с макрокомандами

Получение списка макрокоманд GET http://IP-адрес:порт/префикс/macro/list/

Пример ответа JSON:



```

{
  "macroCommands" : [
    {
      "id" : "04eb71b0-e2e0-445e-ae7a-a036951fb595",
      "name" : "MacroName1"
    },
    {
      "id" : "3fd3bfb0-3a6e-467a-8ff2-88f7b165cf5b",
      "name" : "MacroName2"
    },
    {
      "id" : "941f88d1-b512-4189-84a6-7d274892dd95",
      "name" : "MacroName3"
    }
  ]
}

```

Выполнение макрокоманды GET <http://IP-адрес:порт/префикс/macro/execute/id>

где id - идентификатор из списка макрокоманд.

Возможные коды ошибок при выполнении команд:

- **400** - неправильно сформированный запрос.
- **500** - внутренняя ошибка Сервера.
- **404** - некорректно сформированный идентификатор (только для команды execute)

### 2.1.19 Получение статистики

GET <http://IP-адрес:порт/префикс/statistics/HOSTNAME/DeviceIpint.23/SourceEndpoint.video:0:0> - выдача статистики по указанному источнику видео.

GET <http://IP-адрес:порт/префикс/statistics/webserver> - выдача статистики о работе сервера.

### 2.1.20 Получение данных из системного журнала

GET <http://IP-адрес:порт/префикс/audit/HOST/beginTime/endTime?filter=17-20,6,1:4>

где

- **HOST** - имя Сервера, события которого необходимо получить.
- **beginTime** и **endTime** задают время в формате YYYYMMDDTHHMMSS в часовом поясе UTC+0.
- **filter** - список типов событий, который может быть представлен как диапазоном с разделителями <-> и <->, так и одиночным кодом.

Список типов событий:

- 1 - не публикуется в журнале событий
- 2 - нарушение начала таблицы событий аудита (например, была обрезана вручную)
- 3 - нарушение конца таблицы событий аудита (например, была обрезана вручную)
- 4 - отсутствие части событий аудита (например, были удалены вручную)
- 5 - запись таблицы была изменена
- 6 - простой журнала (отсутствие событий или метки обновления журнала)
- /// Внешние события
- 7 - добавлен пользователь
- 8 - удален пользователь
- 9 - изменены параметры пользователя
- 10 - добавлена роль
- 11 - удалена роль
- 12 - изменены параметры роли
- 13 - вход пользователя
- 14 - выход пользователя
- 15 - добавлено устройство

- 16 - удалено устройство
- 17 - изменены параметры устройства
- 18 - добавлен детектор
- 19 - удален детектор
- 20 - изменены параметры детектора
- 21 - добавлен архив
- 22 - удален архив
- 23 - изменены параметры архива
- 24 - создано правило (макрокоманда) для детектора
- 25 - удалено правило (макрокоманда) для детектора
- 26 - изменены параметры правила (макрокоманды) для детектора
- 27 - инициирована тревога
- 28 - зона поставлена на охрану
- 29 - зона снята с охраны
- 30 - выполнен экспорт из архива
- 31 - добавлен оповещатель (аудио-, email-, sms-)
- 32 - удален оповещатель
- 33 - изменены параметры оповещателя
- 34 - изменен общий параметр
- 35 - изменены параметры записи камеры в архив
- 36 - добавлен агент экспорта
- 37 - удален агент экспорта
- 38 - изменены параметры агента экспорта
- 39 - создана макрокоманда
- 40 - удалена макрокоманда
- 41 - изменены параметры макрокоманды
- 42 - тревога взята пользователем в обработку
- 43 - тревога оценена как опасная
- 44 - тревога оценена как подозрительная
- 45 - тревога оценена как ложная
- 46 - тревога пропущена
- 47 - сервер включен в Аххон-домен
- 48 - сервер исключен из Аххон-домена
- 49 - просмотр архива
- 50 - просмотр камеры
- 51 - просмотр раскладки
- 52 - интеллектуальный поиск в архиве
- 53 - групповой поиск в архиве по лицам
- 54 - групповой поиск в архиве по номерам ТС
- 55 - экспорт системного журнала
- 56 - добавлен каталог LDAP
- 57 - удален каталог LDAP
- 58 - изменены параметры каталога LDAP

Ответ возвращается в JSON формате. Пример ответа:

```

{
  "events": [
    {
      "data": {
        "component": "3.Камера",
        "componentType": "camera",
        "device": "3.Камера",
        "host": "V-SHMELEV",
        "property": "vstream-virtual/folder",
        "setting": "Каталог",
        "value": "D:/Movies/Spirit"
      },
      "eventType": 17,
      "timestamp": "20161205T120410.698000"
    },
    {
      "data": {
        "detector": "Детектор лиц",
        "device": "1.Камера",
        "host": "V-SHMELEV"
      },
      "eventType": 18,
      "timestamp": "20161205T120459.319000"
    }
  ]
}

```

### 2.1.21 Получение информации о загрузке Серверов

GET http://IP-адрес:порт/префикс/statistics/hardware - получение информации о загрузке сети и ЦП указанного Сервера.

GET http://IP-адрес:порт/префикс/statistics/hardware/domain - получение информации о загрузке сети и ЦП всех Серверов Аххон-домена.

Пример ответа:

```
[
  {
    "drives": [
      {
        "capacity": 523920994304,
        "freeSpace": 203887943680,
        "name": "C:\\\\"
      },
      {
        "capacity": 475912990720,
        "freeSpace": 148696813568,
        "name": "D:\\\\"
      },
      {
        "capacity": 0,
        "freeSpace": 0,
        "name": "E:\\\\"
      }
    ],
    "name": "SERVER1",
    "netMaxUsage": "0,006271999999999998",
    "totalCPU": "16,978111368301985"
  }
]
```

### 2.1.22 Получение информации об установленной версии Сервера

GET http://IP-адрес:порт/префикс/product/version

Будет получен ответ следующего вида:

```
{
  "version": "AxxonNext 4.0.2.4483"
}
```

### 2.1.23 Переключение состояния виртуального IP-устройства (HttpListener)

POST http://IP-адрес:порт/device/di/0

с телом {"state": "closed"}

где

- **порт** - порт HttpListener.
- 0/1/2/3 - id датчика.
- **state** - **opened** или **closed**.

Пример:

```
http://127.0.0.1:8080/device/di/0
{"state": "closed"}
```

## 2.2 Клиентское HTTP API

### 2.2.1 Запросы для работы с раскладками и видеостенами

#### 2.2.1.1 Порядок работы

Перед началом работы с HTTP API раскладок и видеостен необходимо в командной строке запустить следующую команду:

```
netsh http add urlacl url=http://IP-адрес:8888/ user=DOMAIN\username
```

где DOMAIN\username относится к Windows (команда **whoami** в командной строке). Порт 8888 должен быть не занят, антивирус и/или фаервол отключены.

#### ⚠ **Внимание!**

Все запросы для работы с раскладками и видеостенами выполняются на Клиенте, который должен быть запущен от имени администратора.  
В запросах необходимо указывать IP-адрес Клиента.

#### ⚠ **Внимание!**

При возникновении ошибок необходимо запустить командную строку от имени администратора и повторно выполнить команду

После успешного выполнения команды можно осуществлять описанные ниже запросы.

#### 2.2.1.2 Получение списка раскладок

GET http://IP-адрес:8888/GetLayouts - получение списка раскладок для текущего пользователя.

Пример ответа:

```
{
  "Description": "",
  "Status": "OK",
  "LayoutInfo": [
    {
      "Id": "102",
      "Name": "Layout name 2"
    },
    {
      "Id": "103",
      "Name": "Layout name 3"
    }
  ]
}
```

Пример сообщения об ошибке:

```
{
  \"result\": \"no layouts\"
}
```

**Примечание**

Ошибка может возникать при запросе списка раскладок с Сервера, на котором включен контроль учетных записей пользователя (UAC). Для устранения ошибки необходимо отключить данную функциональность в системе.

**2.2.1.3 Переключение раскладки на экране**

GET http://IP-адрес:8888/SwitchLayout?layoutId=N&displayId=\\.\\DISPLAY1 - выбор раскладки с id N на мониторе DISPLAY1.

Пример ответа:

```
{
  "Description": "",
  "Status": "OK"
}
```

Пример сообщения об ошибке:

```
{
  \"result\": \"error\"
}
```

**Примечание**

Ошибка может возникать при указании раскладки с несуществующим id

**2.2.1.4 Получение списка видеокамер, отображаемых на раскладке в данный момент**

GET http://IP-адрес:8888/GetCameras?layoutId=N&displayId=\\.\\DISPLAY1 - получение списка видеокамер с раскладки N на мониторе DISPLAY1 для текущего пользователя.

Пример ответа:

```
{
  "Description": "",
  "Status": "OK",
  "CameraInfo": [
    {
      "DisplayName": "1.Camera",
      "Id": "1",
      "Name": "host/HOSTNAME/DeviceIpint1/SourceEndPoint.video:0:0"
    },
    {
      "DisplayName": "2.Camera",
      "Id": "2",
      "Name": " host/HOSTNAME/DeviceIpint2/SourceEndPoint.video:0:0"
    }
  ]
}
```

**Примечание**

Если раскладка с указанным id не будет найдена, то запрос вернет список видеокамер текущей раскладки на указанном мониторе

### 2.2.1.5 Удаление и добавление видеокамер

GET http://IP-адрес:8888/RemoveCamera?displayId=\\.\DISPLAY1&cameraName=Name - удаление видеокамеры с текущей раскладки монитора DISPLAY1.

GET http://IP-адрес:8888/RemoveAllCameras?displayId=\\.\DISPLAY1 - удаление всех видеокамер с текущей раскладки монитора DISPLAY1.

GET http://IP-адрес:8888/AddCamera?displayId=\\.\DISPLAY1&cameraName=Name - добавление видеокамеры на текущую раскладку монитора DISPLAY1.

где Name - имя видеокамеры из ответа на запрос [Получение списка видеокамер, отображаемых на раскладке в данный момент](#).

Пример ответа для всех запросов:

```
{ "Description": "", "Status": "OK" }
```

Пример ошибки для всех запросов:

```
{ "Description": "Error description", "Status": "ERROR" }
```

### 2.2.1.6 Получение списка мониторов

GET http://IP-адрес:8888/GetDisplays - получение списка [мониторов](#) для текущего пользователя.

Пример ответа:

```
{
  "Description": "",
  "Status": "OK",
  "DisplayInfo": [
    {
      "Id": "\\.\DISPLAY1",
      "IsMainForm": true
    },
    {
      "Id": "\\.\DISPLAY2",
      "IsMainForm": false
    }
  ]
}
```

где

**id** - номер монитора;

**IsMainForm** - значение "true" соответствует основному монитору.

Пример сообщения об ошибке:

```
{
  "{ \"result\": \"no displays\" }"
}
```

### 2.2.1.7 Выбор активного монитора

GET http://IP-адрес:8888/SelectDisplay?displayId=\\.\DISPLAY1 - выбор активного [монитора](#).

Пример ответа:

```
{
  "Description": "",
  "Status": "OK"
}
```

Пример сообщения об ошибке:

```
"{\result\": \"error\"}"
```

### 2.2.1.8 Перевод видеокamеры в режим архива

GET http://IP-адрес:8888/GoToArchive?displayId=\\.\\DISPLAY1&cameraName=Name&timestamp=Timestamp

где,

- **displayId** - идентификатор монитора из запроса GetDisplays (см. [Получение списка мониторов](#)).
- **cameraName** - имя видеокamеры, полученное в ответе на запрос GetCameras (см. [Получение списка видеокamер, отображаемых на раскладке в данный момент](#)) или id группы видеокamер.



#### Примечание

Для получения списка групп необходимо использовать следующий запрос

GET http://IP-адрес:8888/GetGroups.

Пример ответа:

Id	"4308f2e2-e57c-4cd0-8a4f-826e8b16d39c"
Name	"Default"

- **timestamp** - время в формате ISO.

Пример запроса:

GET http://localhost:8888/GoToArchive?displayId=\\.\\DISPLAY2&cameraName=hosts/SERVER1/DeviceIpint.1/SourceEndpoint.video:0:0&timestamp=2017-04-07T00:00:00.000

### 2.2.1.9 Переход на раскладку с видеокamерой в режиме погружения

GET http://IP-адрес:8888/GoToImmersion?displayId=\\.\\DISPLAY1&cameraName=Name

где,

- **displayId** - идентификатор монитора из запроса GetDisplays (см. [Получение списка мониторов](#)).
- **cameraName** - имя видеокamеры, полученное в ответе на запрос GetCameras (см. [Получение списка видеокamер, отображаемых на раскладке в данный момент](#)).

### 2.2.1.10 Переход на раскладку с видеокamерой в режиме интеллектуального поиска

GET http://IP-адрес:8888/SearchArchive?displayId=\\.\\DISPLAY1&cameraName=Name&timestamp=Timestamp

где,

- **displayId** - идентификатор монитора из запроса GetDisplays (см. [Получение списка мониторов](#)).
- **cameraName** - имя видеокamеры, полученное в ответе на запрос GetCameras (см. [Получение списка видеокamер, отображаемых на раскладке в данный момент](#)).
- **timestamp** - позиция в архиве в формате ISO.

### 2.2.1.11 Переход к сохраненным результатам интеллектуального поиска в архиве

GET http://IP-адрес:8888/SearchArchive?displayId=\\.\\DISPLAY1&cameraName=Name&query=query\_name

где,

- **displayId** - идентификатор монитора из запроса GetDisplays (см. [Получение списка мониторов](#)).



- **cameraName** - имя видеокамеры, полученное в ответе на запрос GetCameras (см. [Получение списка видеокамер, отображаемых на раскладке в данный момент](#)).
- **query** - имя сохраненного поискового запроса (см. [Сохранение поискового запроса](#)).