

Спецификация

Начиная с версии 6.6.2, система «Линия» позволяет запускать модуль «Наблюдательный пост» и просмотр архива с определенными параметрами командной строки (ниже приведена спецификация, актуальная для 6.9.4 и старше).

Также возможно получение видеопотока и отдельных кадров с применением веб-сервера системы «Линия», что позволяет проводить интеграцию со сторонним программным обеспечением.

Работа с observer.exe из командной строки

Общий формат

```
observer.exe [options] [modes]
```

modes:

```
--simple           ; просмотр в упрощенном режиме
-a, --archive      ; просмотр архива
-m, --allow-multiple-instances ; запуск еще одной копии observer
```

options:

```
-s, --server      default: localhost
-p, --port        default: 9780
-l, --login       default: guard
--pwd            default: ""
-c, --cam-index   default: 0          ; доступно только в --simple режиме
--cam-stream      default: -1         ; доступно только в --simple режиме

-P, --window-pos  coordinates format: X1 Y1 X2 Y2 ; по умолчанию -
                                         предыдущие координаты окна перед закрытием
--tray            ; помещает Наблюдательный пост в трей при старте
-t, --time        date format: YYYY-MM-DD ; по умолчанию - текущая дата
                                         time format: hh:mm:ss ; по умолчанию - текущее время
-h, --help        ; выводит это сообщение
```

Комментарии

Observer можно запускать из командной строки с аргументами. Аргументы делятся на 2 категории:

- переводящие в режим,
- инициализирующие.

Аргументы

Режимы

- --simple - просмотр в упрощенном режиме.
Открытие приложения для просмотра конкретной камеры (возможно, из стороннего приложения)
- -a, --archive - открытие приложения в режиме просмотра архива. Если приложение находится в режиме наблюдения, то произойдет смена режима с последующим подключением к архиву на тот же сервер.

- -m, --allow-multiple-instances - запуск еще одной копии observer'a.

Параметры

- -s, --server [localhost]
Если параметр не задан, то произойдет попытка подключения по адресу 127.0.0.1.
- -p, --port [9780]
Порт для подключения. Если порт не задан, подключение будет осуществляться по стандартному порту 9780.
- -l, --login [guard]
Пользователь по умолчанию - guard (обычно пользователь с самыми ограниченными привилегиями).
- --pwd [""]
Пароль для пользователя. По умолчанию - пустой.
- -c, --cam-index [0]
Параметр доступен только в --simple режиме. При подключении в --simple режиме используются локальные "виды" (а в точности - только одна, указанная камера). По умолчанию идентификатор камеры равен 0.
- --cam-stream [-1]
Параметр доступен только в --simple режиме. При подключении в --simple режиме используются локальные "виды" (а в точности - только одна, указанная камера). По умолчанию идентификатор потока равен -1 (авто), доступные значения: 0 (первый), 1 (второй).
- --window-pos <coordinates>
Положение и размеры окна при запуске.
coordinates: X1 Y1 X2 Y2 - абсолютные координаты окна,
где X1, Y1 - координаты верхнего левого угла, а X2, Y2 - координаты нижнего правого угла.
По умолчанию приложение откроется на том же месте, где было закрыто в предыдущий раз.
- --tray
Помещает «Наблюдательный пост» в трей при старте.
- -t, --time <date> <time>
date format: YYYY-MM-DD; по умолчанию - текущая дата.
time format: hh:mm:ss; по умолчанию - текущее время.
Время для стартового воспроизведения архива.
Если указан этот параметр, но параметр -p (порт) указывает на порт наблюдения (kernel.exe), а не архива (oopnet.exe),
то этот аргумент будет проигнорирован.

Спецификация веб-сервера

- [Спецификация веб-сервера](#)
- [Форматы данных](#)
- [Аутентификация](#)
- [Ресурсы](#)
 - [Камеры](#)
 - [Все камеры](#)
 - [Изображение](#)
 - [Отдельный кадр](#)
 - [М-JPEG поток](#)
 - [H.264 поток](#)
 - [PTZ](#)
 - [Управление PTZ-камерой](#)
 - [Управление фокусом камеры](#)
 - [Работа с экранном меню камеры](#)
 - [Все предустановки PTZ-камеры](#)
 - [Отдельная предустановка PTZ](#)
 - [Предпросмотр изображения с предустановкой PTZ](#)
 - [Микрофоны](#)
 - [Все микрофоны](#)
 - [Аудио](#)
 - [Аудиопоток](#)
 - [OSD \(on-screen display\)](#)
 - [Создание OSD-объекта](#)
 - [Удаление OSD-объекта](#)
 - [Добавление контента в OSD-объект](#)
 - [Удаление контента из OSD-объекта](#)
 - [События](#) (с версии 6.9.4)
 - [Создание события](#)
 - [Изменение события](#)
 - [RPC \(Remote Procedure Call\)](#)
 - [Форматы данных](#)
 - [Время](#)
 - [Таймлайн архива](#)
 - [Структура запроса](#)
 - [Структура ответа](#)
 - [Batch-запросы \(Пакетные запросы\)](#)
 - [Список методов](#)
 - [archive.get_frames_timeline](#)
 - [archive.get_motions_timeline](#)
 - [archive.get_channels_list](#)
 - [archive.get_streams_list](#)
 - [archive.get_frames_list](#)
 - [archive.get_frame](#)
- [Прочее](#)
- [Изменения спецификации](#)

Форматы данных

Поддерживаются JSON или XML (**только** Unix-way перевод строк) представления объектов (MIME-типы `application/json` и `application/xml`). То или иное представление выбирается на основе анализа заголовка `Accept` запроса клиента. Если по заголовку `Accept` невозможно определить желаемое клиентом представление, объекты пересылаются в представлении по умолчанию (XML).

Аутентификация

Осуществляется посредством HTTP Digest Authentication. Имя пользователя и пароль ожидаются в кодировке UTF-8.

Ресурсы

Каждый ресурс описывается следующим набором полей:

URI: абсолютный путь к ресурсу на сервере.

Версия: версия интерфейса, в которой ресурс доступен.

MIME-типы: типы данных, в которых ресурс может быть представлен.

Методы: методы, применимые к ресурсу.

Общее описание ресурса, методов доступа к нему, параметров запросов, форматов данных и т.д.

Камеры

Все камеры

URI: `/cameras`

Версия: 1.2

MIME-типы: `application/json`, `application/xml`

Методы: GET

Список всех доступных на сервере камер. JSON-представление — массив объектов `camera`.

XML-представление:

```
<camera-list>
  <camera/> <!-- opt -->
</camera-list>
```

Отдельная камера в JSON-представлении:

```
{  "uri":  /* string */,
   "name": /* string */,
   "width": /* number (current resolution) */,
   "height": /* number (current resolution) */,
   "image-uri": /* string */,
   "video-uri": /* string */,
   "osd-uri": /* string */,
   "ptz": {
     "pan-tilt": {
       "absolute-uri": /* string */,
       "relative-uri": /* string */
     },
     "zoom": {
       "absolute-uri": /* string */,
```

```

    "relative-uri": /* string */
  },
  "focus": {
    "absolute-uri": /* string */,
    "relative-uri": /* string */
  },
  "menu": {
    "show-uri": /* string */,
    "hide-uri": /* string */,
    "activate-uri": /* string */,
    "move-uri": /* string */
  },
  "presets-uri": /* string */
} }

```

В XML-представлении:

```

<camera>
  <uri>    <!-- xs:anyURI -->      </uri>
  <name>    <!-- xs:string -->      </name>
  <width>    <!-- xs:nonNegativeInteger --> </width>
  <height>    <!-- xs:nonNegativeInteger --> </height>
  <image-uri> <!-- xs:anyURI -->      </image-uri>
  <video-uri> <!-- xs:anyURI -->      </video-uri>
  <osd-uri>   <!-- xs:anyURI -->      </osd-uri>
  <ptz>
    <pan-tilt>
      <absolute-uri> <!-- xs:anyURI -->      </absolute-uri>
      <relative-uri> <!-- xs:anyURI -->      </relative-uri>
    </pan-tilt>
    <zoom>
      <absolute-uri> <!-- xs:anyURI -->      </absolute-uri>
      <relative-uri> <!-- xs:anyURI -->      </relative-uri>
    </zoom>
    <focus>
      <absolute-uri> <!-- xs:anyURI -->      </absolute-uri>
      <relative-uri> <!-- xs:anyURI -->      </relative-uri>
    </focus>
    <menu>
      <activate-uri> <!-- xs:anyURI -->      </activate-uri>
      <show-uri>     <!-- xs:anyURI -->      </show-uri>
      <hide-uri>     <!-- xs:anyURI -->      </hide-uri>
      <move-uri>     <!-- xs:anyURI -->      </move-uri>
    </menu>
    <presets-uri>    <!-- xs:anyURI -->      </presets-uri>
  </ptz>
</camera>

```

Пример работы с ресурсом:

```

GET /cameras HTTP/1.1
Host: 127.0.0.1
Accept: application/json
HTTP/1.1 200 OK
Date: Mon, 23 May 2005 21:07:53 GMT
Content-Type: application/json

```

```

[ { "uri": "http://127.0.0.1/3r5ggsbJHke",
    "name": "Зиповская",
    "width": 640,
    "height": 480,
    "image-uri": "http://127.0.0.1/Ogei7uquahbohyae",
    "video-uri": "http://127.0.0.1/pamua8Eif4moh6ae" },
  { "uri": "http://127.0.0.1/7GheuI95dfghk",
    "name": "Зиповская PTZ",
    "width": 1280,

```

```

"height": 800,
"image-uri": "http://127.0.0.1/PeevupeeNgeir7eu",
"video-uri": "http://127.0.0.1/EK8Shadi3fa3noel",
"osd-uri": "http://127.0.0.1/Ldh485h" },
"ptz": { "focus": { "relative-uri": "http://127.0.0.1/ahgae6Shahloothi" },
  "menu": { "activate-uri": "http://127.0.0.1/quee0tieMelweiRa",
    "show-uri": "http://127.0.0.1/uGai8doosho7Ohfi",
    "hide-uri": "http://127.0.0.1/Fohbureit0eedoo0",
    "move-uri": "http://127.0.0.1/EiY3aach4Nuutha8" },
  "pan-tilt": { "relative-uri": "http://127.0.0.1/isai7chahCuChait" },
  "zoom": { "relative-uri": "http://127.0.0.1/quieJae7zeish3oo" },
  "presets-uri": "http://127.0.0.1/neeloe2polHua3xa" } },
{ "uri": "http://127.0.0.1/hdUneK3oe8",
  "name": "camera1",
  "width": 320,
  "height": 240,
  "image-uri": "http://127.0.0.1/ethieSho5ching4e",
  "video-uri": "http://127.0.0.1/cee9rahtoo4uRooh" },
{ "uri": "http://127.0.0.1/Neo89qli3J",
  "name": "Коридор",
  "width": 320,
  "height": 240,
  "image-uri": "http://127.0.0.1/xohpai8jeQuohm6f",
  "video-uri": "http://127.0.0.1/euDaiheejiaGae3s" }
"osd-uri": "http://127.0.0.1/Oje654fgbd" } ]

```

Изображение

Отдельный кадр

URI: <image-uri объекта Camera>

Версия: 1.0

MIME-типы: image/jpeg, image/png

Методы: GET

Параметры строки запроса:

- **quality** — качество изображения. Задаётся целым числом от 0 (максимальное сжатие) до 100 (максимальное качество). Если не задан явно, используется значение по умолчанию для камеры.
- **resolution** — разрешение изображения. Если не задано, используется текущее разрешение камеры.
- **keep_aspect_ratio** — сохранять оригинальное соотношение сторон: 0 - нет, 1 - да. По умолчанию 0 (нет).

Пример работы с ресурсом:

```

GET /ethieSho5ching4e?keep_aspect_ratio=1&resolution=640x480 HTTP/1.1
Host: 127.0.0.1
Accept: image/*
HTTP/1.1 200 OK
Date: Mon, 23 May 2005 23:15:27 GMT
Content-Type: image/jpeg
Content-Length: 23185

```

[Данные изображения]

М-JPEG поток

URI: <video-uri объекта Camera>

Версия: 1.0

MIME-типы: multipart/x-mixed-replace, image/jpeg

Методы: GET

Параметры строки запроса:

- **fps** — максимальное количество кадров в секунду. Если параметр не задан явно, используются текущие настройки камеры.
- **quality** — качество изображения. Задаётся целым числом от 0 (максимальное сжатие) до 100 (максимальное качество). Если параметр не задан, используется значение для камеры по умолчанию.
- **resolution** — разрешение изображения. Если не задано, используется текущее разрешение камеры.
- **keep_aspect_ratio** — сохранять оригинальное соотношение сторон: 0 - нет, 1 - да. По умолчанию 0 (нет).

Пример работы с ресурсом:

```
GET /cee9rahtoo4uRooh?keep_aspect_ratio=0&fps=5&resolution=320x240 HTTP/1.1
```

```
Host: 127.0.0.1
```

```
HTTP/1.1 200 OK
```

```
Date: Mon, 23 May 2005 23:15:27 GMT
```

```
Content-Type: multipart/x-mixed-replace; boundary=xyzzy
```

```
--xyzzy
```

```
Content-Type: image/jpeg
```

```
Content-Length: 15382
```

```
[Данные кадра 1]
```

```
--xyzzy
```

```
Content-Type: image/jpeg
```

```
Content-Length: 16105
```

```
[Данные кадра 2]
```

```
[и т.д.]
```

Н.264 поток

URI: <streaming-uri объекта Camera>

Методы: GET

Для формирования ссылки на поток следует использовать значение и идентификатор потока, который составляется из имени и расширения. На данный момент возможны следующие имена:

- **sub** - второй поток (низкое разрешение);
- **main** - первый поток (высокое разрешение).

и расширения:

- **flv** - Flash Video;
- **m3u8** - HTTP Live Streaming (HLS) playlist.

Примеры ссылок на потоки, где /kfd3ado1sdrms - uri объекта Camera:

/kfd3adol1sdrms/streaming/main.flv - первый поток в формате Flash Video;
/kfd3adol1sdrms/streaming/sub.m3u8 - второй поток в формате HLS;
/kfd3adol1sdrms/streaming/sub.flv - второй поток в формате Flash Video;
/kfd3adol1sdrms/streaming/main.m3u8 - первый поток в формате HLS.

На клиентах, где невозможно авторизовать запрос стандартными средствами (HTTP Digest/Basic Authentication), возможна передача заголовка Authorization одним из параметров запроса, например /kfd3adol1sdrms/streaming/main.flv?authorization=Basic%20d2ViOg==

PTZ

Управление PTZ-камерой

URI: <absolute-uri или relative-uri объектов Camera/ptz/pan-tilt и Camera/ptz/zoom>

Версия: 1.0

MIME-типы: application/json, application/xml

Методы: PUT

Управление PTZ-камерой производится путём записи по указанным URI объектов pan-tilt-data и zoom-data.

JSON-представление pan-tilt-data

```
{ "x": /* number */,  
  "y": /* number */,  
  "speed": /* number */ }
```

XML-представление:

```
<pan-tilt-data>  
  <x>      <!-- xs:float --> </x>  
  <y>      <!-- xs:float --> </y>  
  <speed> <!-- xs:float --> </speed>  
</pan-tilt-data>
```

JSON-представление zoom-data:

```
{ "zoom": /* number */,  
  "speed": /* number */ }
```

XML-представление:

```
<zoom-data>  
  <zoom> <!-- xs:float --> </zoom>  
  <speed> <!-- xs:float --> </speed>  
</zoom-data>
```

Допустимые значения параметра speed в интервале [0, 1]. Параметры x, y и zoom для команд относительного поворота задаются в абстрактных единицах, на текущий момент допустимы только значения -1 и +1.

Пример — поворот относительно текущего положения:

```
PUT /isai7chahCuChait HTTP/1.1  
Host: 127.0.0.1  
Content-Type: application/xml
```

```
<?xml version="1.0" encoding="UTF-8"?>  
<pan-tilt-data>
```



```
<x>-1.0</x>
<y>0.0</y>
<speed>1.0</speed>
</pan-tilt-data>
HTTP/1.1 204 No Content
Date: Tue, 24 May 2005 17:21:15 GMT
```

Пример — относительное масштабирование:

```
PUT /quieJae7zeish3oo HTTP/1.1
Host: 127.0.0.1
Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8"?>
<zoom-data>
  <zoom>1.0</zoom>
  <speed>1.0</speed>
</zoom-data>
HTTP/1.1 204 No Content
Date: Tue, 24 May 2005 17:21:37 GMT
```

Управление фокусом камеры

URI: <absolute-uri или relative-uri объекта Camera/ptz/focus>

Версия: 1.0

MIME-типы: application/json, application/xml

Методы: PUT

JSON-представление данных фокуса:

```
{ "focus": /* number */,
  "speed": /* number */ }
```

XML-представление:

```
<focus-data>
  <focus> <!-- xs:float --> </focus>
  <speed> <!-- xs:float --> </speed>
</focus-data>
```

Допустимые значения параметра `speed` в интервале $[0, 1]$. Параметр `focus` для команд относительного изменения фокуса задаются в абстрактных единицах, на текущий момент допустимы только значения -1 и $+1$.

Пример работы с ресурсом:

```
PUT /ahgae6Shahlouthi HTTP/1.1
Host: 127.0.0.1
Content-Type: application/json

{ "focus": -1.0,
  "speed": 1.0 }
HTTP/1.1 204 No Content
Date: Tue, 24 May 2005 17:38:27 GMT
```

Работа с экраным меню камеры

URI: <activate-uri, show-uri, hide-uri и move-uri объекта Camera/ptz/menu>

Версия: 1.0

MIME-типы: application/json, application/xml

Методы: PUT

Ресурсы представляют собой команды, запускаемые методом PUT. В ресурсы по `activate-uri` (выбрать пункт меню),

`show-uri` (отобразить меню) и `hide-uri` (скрыть меню) допускается запись сообщений без тела запроса.

Перемещение по пунктам меню осуществляется путём записи команды меню в ресурс по `move-uri`.

JSON-представление команды меню:

```
{ "action": /* string */ }
```

XML-представление:

```
<menu-command action="xs:string"/>
```

Где для `action` допустимо одно из значений: `left`, `right`, `up` или `down`.

Пример работы с меню — отобразить меню:

```
PUT /uGai8doosho7Ohfi HTTP/1.1
Host: 127.0.0.1
HTTP/1.1 204 No Content
Date: Tue, 24 May 2005 13:28:15 GMT
```

Переход в меню к следующему ниже пункту:

```
PUT /EiY3aach4Nuutha8 HTTP/1.1
Host: 127.0.0.1
Content-Type: application/json

{ "action": "down" }
HTTP/1.1 204 No Content
Date: Tue, 24 May 2005 13:28:19 GMT
```

Выбрать пункт меню:

```
PUT /quee0tieMelweiRa HTTP/1.1
Host: 127.0.0.1
HTTP/1.1 204 No Content
Date: Tue, 24 May 2005 13:28:22 GMT
```

Скрыть меню:

```
PUT /Fohbureit0eedoo0 HTTP/1.1
Host: 127.0.0.1
HTTP/1.1 204 No Content
Date: Tue, 24 May 2005 13:28:24 GMT
```

Все предустановки PTZ камеры

URI: `<presets-uri объекта Camera>`

Версия: 1.0

MIME-типы: `application/json`, `application/xml`

Методы: GET, POST

Список предустановок PTZ конкретной камеры.

Пример работы с ресурсом:

```
GET /neeloe2polHua3xa HTTP/1.1
Host: 127.0.0.1
Accept: application/xml
HTTP/1.1 200 OK
Date: Tue, 24 May 2005 17:14:16 GMT
Content-Type: application/xml
```

```
<?xml version="1.0" encoding="UTF-8"?>
<ptz-preset-list>
  <ptz-preset>
    <name>Unu</name>
    <activate-uri>http://127.0.0.1/OhX7iet9aithoofu</activate-uri>
    <preset-uri>http://127.0.0.1/doo3reelaef8Vudo</preset-uri>
    <preview-uri>http://127.0.0.1/eer7vei9thoa4Eik</preview-uri>
    <update-uri>http://127.0.0.1/keenoo3IthoP4zai</update-uri>
  </ptz-preset>
  <ptz-preset>
    <name>Du</name>
    <activate-uri>http://127.0.0.1/xee9uz7Aich4tuga</activate-uri>
    <preset-uri>http://127.0.0.1/gool9hieR3iu5hoh</preset-uri>
    <preview-uri>http://127.0.0.1/ingae4ooCh8ahb4a</preview-uri>
    <update-uri>http://127.0.0.1/ohxuof4Su7ekiuDi</update-uri>
  </ptz-preset>
  <ptz-preset>
    <name>Tri</name>
    <activate-uri>http://127.0.0.1/Uosuu4aivuuchei4</activate-uri>
    <preset-uri>http://127.0.0.1/AtejaelbaeD6Ku4A</preset-uri>
    <preview-uri>http://127.0.0.1/Sheip2ooXohd1Phe</preview-uri>
    <update-uri>http://127.0.0.1/haurieP9chosaif4</update-uri>
  </ptz-preset>
</ptz-preset-list>
```

```
POST /neeloe2polHua3xa HTTP/1.1
Host: 127.0.0.1
Accept: application/xml
Content-Type: application/xml
```

```
<?xml version="1.0" encoding="utf-8"?>
<ptz-preset>
  <name>Kvar</name>
</ptz-preset>
HTTP/1.1 201 Created
Date: Tue, 24 May 2005 17:14:23 GMT
Content-Type: application/xml
```

```
<?xml version="1.0" encoding="utf-8"?>
<ptz-preset>
  <name>Kvar</name>
  <activate-uri>http://127.0.0.1/owaSiewahch3ohch</activate-uri>
  <preset-uri>http://127.0.0.1/xie4Aoz8aqu0aigh</preset-uri>
  <preview-uri>http://127.0.0.1/laeMeef4Kariphu6</preview-uri>
  <update-uri>http://127.0.0.1/ahghohK4iezoovah</update-uri>
</ptz-preset>
```

Отдельная предустановка PTZ

URI: <preset-uri объекта PTZPreset>

Версия: 1.0

MIME-типы: application/json, application/xml

Методы: DELETE, GET, PUT

JSON-представление:

```
{ "name": /* string */,
  "activate-uri": /* string */,
  "preview-uri": /* string */,
```

```
"update-uri": /* string */ }
```

XML-представление:

```
<ptz-preset>
  <name>                                <!-- xs:string --> </name>
  <activate-uri> <!-- xs:anyURI --> </activate-uri>
  <preview-uri>  <!-- xs:anyURI --> </preview-uri>
  <update-uri>   <!-- xs:anyURI --> </update-uri>
</ptz-preset>
```

Пример работы с ресурсом:

```
GET /phiwohGh5aer0ye4 HTTP/1.1
Host: 127.0.0.1
Accept: application/json
HTTP/1.1 200 OK
Date: Tue, 24 May 2005 14:39:18 GMT
Content-Type: application/json
```

```
{ "name": "Xyzzy!",
  "activate-uri": "http://127.0.0.1/ju9moT7wuciiW7ae",
  "preview-uri": "http://127.0.0.1/eite4ohLiQu5Veoc",
  "update-uri": "http://127.0.0.1/WieB4ahng4ieghi9" }
PUT /phiwohGh5aer0ye4 HTTP/1.1
Host: 127.0.0.1
Accept: application/xml
Content-Type: application/json
```

```
{ "name": "Zyxxxy?",
  "activate-uri": "http://127.0.0.1/ju9moT7wuciiW7ae",
  "preview-uri": "http://127.0.0.1/eite4ohLiQu5Veoc",
  "update-uri": "http://127.0.0.1/WieB4ahng4ieghi9" }
HTTP/1.1 200 OK
Date: Tue, 24 May 2005 14:39:37 GMT
Content-Type: application/xml
```

```
<?xml version="1.0" encoding="utf-8"?>
<ptz-preset>
  <name>Zyxxxy?</name>
  <activate-uri>http://127.0.0.1/ju9moT7wuciiW7ae</activate-uri>
  <preset-uri>http://127.0.0.1/phiwohGh5aer0ye4</preset-uri>
  <preview-uri>http://127.0.0.1/eite4ohLiQu5Veoc</preview-uri>
  <update-uri>http://127.0.0.1/WieB4ahng4ieghi9</update-uri>
</ptz-preset>
```

Предпросмотр изображения с предустановкой PTZ

URI: <preview-uri объекта PTZPreset>

Версия: 1.0

MIME-типы: image/jpeg, image/png

Методы: GET

Размер изображения предварительного просмотра зависит от камеры.

Пример работы с ресурсом:

```
GET /eite4ohLiQu5Veoc HTTP/1.1
Host: 127.0.0.1
Accept: image/png
HTTP/1.1 200 OK
Date: Tue, 24 May 2005 11:27:16 GMT
```

Content-Type: image/png
Content-Length: 2108

[Данные изображения]

Микрофоны

Все микрофоны

URI: /microphones

Версия: 1.2

MIME-типы: application/json, application/xml

Методы: GET

Список всех доступных на сервере микрофонов.

JSON-представление — массив объектов microphone.

XML-представление:

```
<microphone-list>
  <microphone/> <!-- opt -->
</microphone-list>
```

Отдельный микрофон в JSON-представлении:

```
{ "uri": /* string */,
  "name": /* string */,
  "audio-uri": /* string */,
  "camera-uri": /* string */ }
```

В XML-представлении:

```
<microphone>
  <uri>                                <!-- xs:anyURI --> </uri>
  <name>                                <!-- xs:string --> </name>
  <audio-uri>                          <!-- xs:anyURI --> </audio-uri>
  <camera-uri>                         <!-- xs:anyURI --> </camera-uri>
</microphone>
```

Пример работы с ресурсом:

```
GET /microphones HTTP/1.1
Host: 127.0.0.1
Accept: application/json
HTTP/1.1 200 OK
Date: Mon, 23 May 2005 21:07:53 GMT
Content-Type: application/json
```

```
[ { "uri": "http://127.0.0.1/6Hjk9l4dudlf",
    "name": "Зиповская",
    "audio-uri": "http://127.0.0.1/Ogei7uquahbohvae",
    "camera-uri": "http://127.0.0.1/pamua8Eif4moh6ae" },
  { "uri": "http://127.0.0.1/Pqja954nfrdqcb",
    "name": "Microphone1",
    "audio-uri": "http://127.0.0.1/ethieSho5ching4e",
    "camera-uri": "http://127.0.0.1/cee9rahtoo4uRooh" },
  { "uri": "http://127.0.0.1/Q57mak8maogk",
    "name": "Коридор",
    "audio-uri": "http://127.0.0.1/xohpai8jeQuohm6f",
```

```
"camera-uri": "http://127.0.0.1/euDaiheejaGae3s" } ]
```

Аудио

Аудиопоток

URI: <audio-uri объекта Microphone>

Версия: 1.1

MIME-типы: audio/x-wav, audio/mpeg

Методы: GET

Параметры строки запроса:

- **sample_rate** — частота выборки сигнала. Может принимать значения [8000, 11025, 12000, 16000, 22050, 24000, 32000, 44100, 48000] Hz. Если не задан явно, используется значение 22050.
- **bit_rate** — битрейт (например, 128).
Если не задан, используется 64 kbps для audio/mpeg и 22050 * 16 bps (22.05 kHz * 16 bit) для audio/x-wav, соответственно.

Пример работы с ресурсом:

```
GET /ethieSho5ching4e?sample_rate=44100&bit_rate=128 HTTP/1.1
Host: 127.0.0.1
Accept: audio/mpeg
HTTP/1.1 200 OK
Date: Mon, 23 May 2005 23:15:27 GMT
Content-Type: audio/mpeg
```

[Аудиоданные]

Трансляция аудио длится до разрыва соединения любой стороной.

OSD (on-screen display)

Создание OSD-объекта

URI: <osd-uri объекта Camera>

Версия: 1.5

MIME-типы: application/json, application/xml

Методы: POST

Тело запроса в JSON-представлении:

```
{ "name": /* string */,
  "left": /* number */,
  "top": /* number */,
  "font-size": /* number */,
  "font-color": /* string */,
  "line-count": /* number */,
  "draw-background": /* number */ }
```

В XML-представлении:

```
<osd-zone-data>
  <name>                                <!-- xs:string -->                                </name>
```

```
<left>          <!-- xs:nonNegativeInteger --> </left>
<top>           <!-- xs:nonNegativeInteger --> </top>
<font-size>     <!-- xs:nonNegativeInteger --> </font-size>
<font-color>    <!-- xs:string --> </font-color>
<line-count>    <!-- xs:nonNegativeInteger --> </line-count>
<draw-background> <!-- xs:nonNegativeInteger --> </draw-background>
</osd-zone-data>
```

Координаты вывода и размер шрифта задаются в % относительно размеров кадра.

Параметр **<font-color>** задается строкой вида **#RGB**, где R, G, B - значения цветовых составляющих в hex-формате, в диапазоне [0..255].

Пример: #FF0000, #0A5B4F.

Параметр **<draw-background>** задает выделение заднего фона текста (затемнение или осветление в зависимости от яркостной составляющей цвета текста): 0 - не выделять, 1 - выделять.

Ответ сервера в случае успешного выполнения запроса:

```
HTTP/1.1 201 Created
Location: <URI OSD-объекта>
```

Пример работы с ресурсом:

```
POST <osd-uri объекта Camera> HTTP/1.1
Host: 127.0.0.1
Content-Type: application/json
```

```
{ "name": "xyz",
  "left": 10,
  "top": 20,
  "font-size": 10,
  "font-color": "#00FF00",
  "line-count": 10
  "draw-background": 1 }
HTTP/1.1 201 Created
Date: Mon, 23 May 2005 23:15:27 GMT
Location: <osd-uri объекта Camera>/xyz
```

Удаление OSD-объекта

URI: <URI OSD-объекта>

Версия: 1.2

Методы: DELETE

Пример работы с ресурсом:

```
DELETE <URI OSD-объекта> HTTP/1.1
Host: 127.0.0.1
HTTP/1.1 204 No content
Date: Mon, 23 May 2005 23:15:27 GMT
```

Добавление контента в OSD-объект

URI: <URI OSD-объекта>

Версия: 1.2

MIME-типы: text/plain

Методы: POST

Пример работы с ресурсом:

```
POST <URI OSD-объекта> HTTP/1.1
```

Host: 127.0.0.1
Content-Type: text/plain
Content-Length: 15

Hello, world!!!
HTTP/1.1 201 Created
Date: Mon, 23 May 2005 23:15:27 GMT

Удаление контента из OSD-объекта

URI: <URI OSD-объекта>/clear

Версия: 1.2

MIME-типы: text/plain

Методы: PUT

Пример работы с ресурсом:

PUT <URI OSD-объекта>/clear HTTP/1.1
Host: 127.0.0.1
HTTP/1.1 204 No content
Date: Mon, 23 May 2005 23:15:27 GMT

События

Создание события

URI: /events

MIME-типы: application/json, application/xml

Методы: POST

Тело запроса в JSON-представлении:

```
{ "time" : /*string*/,  
  "endtime" : /*string*/,  
  "source" : /*string*/,  
  "name" : /*string*/,  
  "device": /*number*/,  
  "data" : /*string*/ }
```

Значение полей:

time - время начала события в формате ISO 8601. Необязательный параметр. Значение по умолчанию - текущее время сервера.

endtime - время окончания длительного события в формате ISO 8601. Необязательный параметр. Значение по умолчанию - нет (событие мгновенное). Допускается указание строки "0" как указания того, что событие - длительное, но незавершенное на данный момент.

source - источник события. Необязательный параметр. Значение по умолчанию - пустая строка.

name - название события. Необязательный параметр. Значение по умолчанию - пустая строка.

device - порядковый номер камеры. Необязательный параметр. Значение по умолчанию - -1 (событие не привязано к камере).

data - данные события. Необязательный параметр. Значение по умолчанию - пустая строка.

Ответ сервера в случае успешного выполнения запроса:

HTTP/1.1 201 Created
Location: <URI объекта события>

Пример работы с ресурсом:


```
POST /events HTTP/1.1
Host: localhost:9786
Content-Type: application/json
```

```
{
  "time" : "2005-08-09T18:31:42.201",
  "source" : "test source",
  "name" : "test name",
  "device": 0,
  "data" : "test data"
}
```

```
HTTP/1.1 201 Created
Date: Mon, 23 May 2009 23:15:27 GMT
Location: events/gds2w8sd1w
```

Заголовок Location содержит адрес созданного ресурса события. По нему в дальнейшем можно получить доступ к объекту события.

Изменение события

URI: <URI объекта события>

MIME-типы: application/json, application/xml

Методы: PUT

В теле запроса допускается использование любых полей, перечисленных в разделе «Создание события».

Пример работы с ресурсом:

```
PUT /events/gds2w8sd1w HTTP/1.1
Host: localhost:9786
Content-Type: application/json
```

```
{
  "endtime" : "2005-08-09T18:31:42.201"
}
```

```
HTTP/1.1 204 No content
Date: Mon, 23 May 2005 23:15:27 GMT
```

RPC (Remote Procedure Call)

URI: /rpc

MIME-типы: application/json, application/x-msgpack

Методы: POST

Ресурс предоставляет простой доступ к возможностям полноценного протокола работы с сервером, используемого, например, приложением «Наблюдательный пост», без необходимости реализовывать сетевую часть взаимодействия, позволяя использовать в качестве транспорта HTTP

Форматы данных

Данные запроса могут быть переданы в JSON (версия сервера 7.1.1 и выше) или MessagePack (версия 7.0 и выше) форматах. Сервер всегда отдаёт ответ в формате запроса. Кодировка текста - UTF-8. Для достижения максимальной производительности рекомендуется использовать MessagePack. Все примеры будут приведены в формате JSON.

Время

Время представляется в виде массива целых чисел, элементы которого соответствуют году, месяцу, дню, часу, минуте, секунде и миллисекунде. Незначащие завершающие элементы могут быть опущены, например [2016, 11, 30, 15] соответствует 30 октября 2016 года 15:00 и может использоваться вместо [2016, 11, 30, 15, 0, 0, 0].

Таймлайн архива

Здесь и далее под таймлайном понимается битовое представление информации о наличии записи в архиве: есть ли архив в единицу времени или нет. В таком определении единица времени выступает разрешением таймлайна. Сервер даёт возможность получать таймлайны с любым разрешением до 1 секунды. Разрешение не обязано быть кратным 1 секунде, минуте, дню и т.д. Любой таймлайн может быть вычислен с точность, например, до 10 секунд, 16 секунд, 2 минут, 30 минут, 7 дней и т.д.

Примеры таймлайнов:

- [0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] - таймлайн с 1 января 2016 года по 1 января 2017 года с разрешением 1 месяц. В данном случае видно, что на сервере доступен архив за февраль и март;
- [1, 0] - таймлайн с 1 января 2016 года по 1 января 2017 года с разрешением 6 месяцев. Доступен архив за первое полугодие 2016 года;
- [0, 1, 0] - таймлайн с 1 января 2015 года по 1 января 2018 года с разрешением 1 год. Доступен архив за 2016 год;
- [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] - таймлайн за февраль 2016 года. Доступен архив за 5 и 17 февраля. Следует помнить, что при построении любых таймлайнов количество дней любого месяца любого года принято за 31, т.е при запросе, например, таймлайна за май, июнь и июль сервер вернёт массив из 93 элементов, где элемент, соответствующий 31 июню всегда будет равен нулю, т.к. такого дня не существует.
- [0, 0, 0, 0, 0, 0, 1, 1, 0, 0] - таймлайн с 00:30 часов 10 марта 2016 года по 00:41 часов 10 марта 2016 года. Доступен архив за 00:36 и 00:37.

Структура запроса

В случае одиночного запроса, он представляет собой словарь с одним обязательным полем:

- `method` - имя метода.

и необязательными полями:

- params - словарь с параметрами;
- id - целое число, идентификатор запроса, который будет включен в ответ;

Пример запроса:

```
1{
2  "id" : 42,
3  "method" : "some_method",
4  "params" :
5  {
6    "some_param" : "some_value"
7  }
8}
```

Структура ответа

Ответ также является словарём, набор полей которого отличается в зависимости от успешности обработки запроса:

- `id` - целое число, идентификатор запроса, который был передан в запросе;
- `result` - словарь с результатом; содержится в ответе только в случае успешной обработки команды;
- `error` - словарь с информацией о возникшей ошибке.

Пример ответа в случае успешного выполнения обработки запроса:

```
POST /rpc HTTP/1.1
Host: 127.0.0.1
Content-Type: application/json

{
  "method" : "sum_two_numbers",
  "params" :
  {
    "first" : 30,
    "second" : 12
  }
}
HTTP/1.1 200 OK
Date: Tue, 07 Jun 2016 07:52:46 GMT
Content-Type: application/json

{
  "result" :
  {
    "sum" : 42
  }
}
```

Пример ответа в случае возникновения ошибки во время обработки запроса:

```
POST /rpc HTTP/1.1
Host: 127.0.0.1
Content-Type: application/json

{
  "method" : "sum_two_numbers",
  "params" :
  {
    "second" : 30
  }
}
HTTP/1.1 200 OK
Date: Tue, 07 Jun 2016 07:52:46 GMT
Content-Type: application/json

{
  "error" :
  {
    "type" : "invalid_param",
    "message" : "first"
  }
}
```

Batch-запросы (Пакетные запросы)

Сервер поддерживает выполнение списка команд, переданных одновременно. В этом случае команды должны быть представлены в виде массива отдельных запросов. Ответ на такой запрос

будет представлен в виде массива отдельных ответов, порядок которых будет соответствовать порядку запросов:

```
POST /rpc HTTP/1.1
Host: 127.0.0.1
Content-Type: application/json

[
  {
    "method" : "mul_two_numbers",
    "params" :
    {
      "first" : 7,
      "second" : 6
    }
  },
  {
    "method" : "sub_two_numbers",
    "params" :
    {
      "first" : 30
    }
  }
]
HTTP/1.1 200 OK
Date: Tue, 07 Jun 2016 07:52:46 GMT
Content-Type: application/json
```

```
[
  {
    "result" :
    {
      "mul" : 42
    }
  },
  {
    "error" :
    {
      "type" : "invalid_param",
      "message" : "second"
    }
  }
]
```

Список методов

archive.get_frames_timeline

Получение таймлайна.

Параметры запроса:

- **channel** - необязательный идентификатор канала, если не указан, используются все доступные;
- **stream** - идентификатор потока ("video", "video2", "video3", "audio");
- **start_time** - время начала таймлайна;
- **end_time** - время конца таймлайна;
- **unit_len** - разрешение таймлайна в секундах.

Содержимое ответа:

- **timeline** - запрошенный таймлайн (описание представления см. в "Форматы данных").

Пример запроса списка дней с архивом любой камеры за февраль 2016 года:

```
1{
2  "method" : "archive.get_frames_timeline",
3  "params" :
4  {
5    "start_time" : [2016, 2, 1],
6    "end_time" : [2016, 3, 1],
7    "unit_len" : 86400 //количество секунд в одном дне - 60 * 60 * 24
8  }
9}
```

Пример ответа:

```
1{
2  "result" :
3  {
4    "timeline" : [0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
5  }
6}
```

Пример запроса списка минут с архивом второго потока камеры 0 за 1 февраль 2016 года:

```
1{
2  "method" : "archive.get_frames_timeline",
3  "params" :
4  {
5    "channel" : 0,
6    "stream" : "video2",
7    "start_time" : [2016, 2, 1],
8    "end_time" : [2016, 2, 2],
9    "unit_len" : 60
10  }
11}
```

Пример ответа:

```
1{
2  "result" :
3  {
4    "timeline" : [0, 0, 1, ... , 0, 0, 1] //массив из 1440 элементов (кол-во минут в
дне)
5  }
6}
```

archive.get_motions_timeline

Получение таймлайна детекций движения по заданному фильтру (аналитика в архиве).

Параметры запроса:

- channel - необязательный идентификатор канала, если не указан, используются все доступные;
- start_time - время начала таймлайна;
- end_time - время конца таймлайна (общая продолжительность временного отрезка не должна превышать 5 минут);
- unit_len - разрешение таймлайна в секундах;
- filter - словарь, задающий параметры фильтрации.

Параметры фильтрации (все параметры необязательны):

- `min_area` - минимальный размер движения в виде числа с плавающей точкой от 0.0 до 1.0, например 0.1 для движения размером не менее 10% от размера кадра;
- `max_area` - максимальный размер движения;
- `mask64` - 64-битная маска по сетке 8x8 ячеек, которая накладывается на всё изображение камеры, представленная в виде строки из 64 символов, каждый из которых может быть "0" или "1", например
"00000000000100" для фильтрации по движению, затрагивающему четвертую ячейку второй строки изображения, представленного в виде таблицы 8x8 ячеек;
- `colors` - массив, где каждый элемент представляет из себя массив из 3-х чисел, соответствующих RGB-компонентам искомого цвета в диапазоне [0, 255].

Содержимое ответа:

- `timeline` - запрошенный таймлайн (описание представления см. в "Форматы данных").

Пример запроса:

```
1{
2  "method" : "archive.get_motions_timeline",
3  "params" :
4  {
5    "start_time" : [2016, 2, 1, 18, 0, 0],
6    "end_time" : [2016, 2, 1, 18, 5, 0],
7    "unit_len" : 60,
8    "filter" :
9    {
10      "min_area" : 0.03,
11      "colors" : [[120, 16, 23], [43, 55, 233]]
12    }
13  }
14}
```

Пример ответа:

```
1{
2  "result" :
3  {
4    "timeline" : [0, 1, 0, 0, 1]
5  }
6}
```

`archive.get_channels_list`

Получение списка камер, доступных в архиве, за указанный отрезок времени.

Параметры запроса:

- `start_time` - начало временного отрезка для поиска доступных каналов;
- `end_time` - конец временного отрезка для поиска доступных каналов.

Содержимое ответа:

- `channels` - массив, где каждый элемент соответствует одному каналу.

Содержимое элемента массива "channels":

- `channel` - идентификатор канала.

Пример запроса:

```
1{
2  "method" : "archive.get_channels_list",
3  "params" :
4  {
5    "start_time" : [2016, 1, 6],
6    "end_time" : [2016, 1, 7]
7  }
8}
```

Пример ответа:

```
1{
2  "result" :
3  {
4    "channels" :
5    [
6      {
7        "channel" : 0
8      },
9      {
10       "channel" : 1
11     }
12   ]
13 }
14}
```

archive.get_streams_list

Получение списка потоков конкретной камеры, доступных в архиве, за указанный отрезок времени.

Параметры запроса:

- channel - идентификатор канала;
- start_time - начало временного отрезка для поиска доступных потоков;
- end_time - конец временного отрезка для поиска доступных потоков.

Содержимое ответа:

- streams - массив, где каждый элемент соответствует одному потоку.

Содержимое элемента массива "streams":

- stream - идентификатор потока.

Пример запроса:

```
1{
2  "method" : "archive.get_streams_list",
3  "params" :
4  {
5    "channel" : 0,
6    "start_time" : [2016, 1, 6],
7    "end_time" : [2016, 1, 7]
8  }
9}
```

Пример ответа:

```
1{
```

```

2  "result" :
3  {
4      "streams" :
5      [
6          {
7              "stream" : "video"
8          },
9          {
10             "stream" : "video2"
11          },
12          {
13             "stream" : "video3"
14          },
15          {
16             "stream" : "audio"
17          },
18      ]
19  }
20}

```

archive.get_frames_list

Получение списка кадров, доступных для загрузки.

Параметры запроса:

- channel - идентификатор канала;
- stream - идентификатор потока;
- start_time - начало временного отрезка для поиска кадров;
- end_time - конец временного отрезка для поиска кадров.

Содержимое ответа:

- frames_list - массив, где каждый элемент соответствует одному кадру.

Содержимое элемента массива "frames_list":

- id - идентификатор кадра;
- gop_index - порядковый номер кадра в GOP, 0 - опорный/контрольный/intra кадр;
- timestamp - время кадра.

Пример запроса списка кадров потока "video" камеры 0 за 13:00 часов 1 марта 2016 года:

```

1{
2  "method" : "archive.get_frames_list",
3  "params" :
4  {
5      "channel" : 0,
6      "stream" : "video",
7      "start_time" : [2016, 3, 1, 13, 0],
8      "end_time" : [2016, 3, 1, 14, 0]
9  }
10}

```

Пример ответа:

```

1{
2  "result" :
3  {
4      "frames_list" :
5      [

```



```

6      {
7          "id" : "f9lk3jfs42df41fsdj4",
8          "gop_index" : 12,
9          "timestamp" : [2016, 3, 1, 13, 11, 43, 543]
10     },
11     {
12         "id" : "kdj48dh2n",
13         "gop_index" : 13,
14         "timestamp" : [2016, 3, 1, 13, 11, 43, 659]
15     },
16     ... ,
17     {
18         "id" : "ja4wds94dk34",
19         "gop_index" : 2,
20         "timestamp" : [2016, 3, 1, 13, 58, 12, 78]
21     }
22 ]
23 }
24 }

```

archive.get_frame

Получение данных конкретного кадра. Метод поддерживается только в формате application/x-msgpack.

Параметры запроса:

- channel - идентификатор канала;
- stream - идентификатор потока;
- id - идентификатор кадра.

Содержимое ответа:

- frame - словарь с метаинформацией и данными кадра.

Содержимое словаря "frame":

- info - словарь с информацией о кадре;
- raw_bytes - данные кадра.

Содержимое словаря "info" при запросе видео кадра:

- timestamp - время кадра;
- gop_index - порядковый номер кадра в GOP;
- width (может отсутствовать) - ширина кадра;
- height (может отсутствовать) - высота кадра;
- codec - одно из значений: "h264", "mpeg4", "mjpeg".

Содержимое словаря "info" при запросе аудио кадра:

- timestamp - время кадра;
- codec - на данный момент возможно только значение "pcm";
- channels - количество каналов звука;
- sps - количество отчетов в секунду (samples per second);
- bps - количество бит в секунду (bit per second).

Пример запроса:

```
1 {
```

```
2  "method" : "archive.get_frame",
3  "params" :
4  {
5    "channel" : 0,
6    "stream" : "video",
7    "id" : "deadbeef"
8  }
9}
```

Пример ответа:

```
1{
2  "result" :
3  {
4    "frame" :
5    {
6      "info" :
7      {
8        "timestamp" : [2016, 12, 23, 12, 54, 26, 943],
9        "gop_index" : 21,
10       "codec" : "h264"
11      },
12      "raw_bytes" : ...
13    }
14  }
15}
```

Прочее

Версия протокола

URI: /version

Версия: 1.1

MIME-типы: application/json, application/xml

Методы: GET

Версия протокола, поддерживаемая данным сервером. Применим только метод GET без параметров строки запроса.

Версия протокола в JSON-представлении выглядит так:

```
{ "major": /* number */,
  "minor": /* number */ }
```

XML-представлении так:

```
<version>
  <major> <!-- xs:int --> </major>
  <minor> <!-- xs:int --> </minor>
</version>
```

Пример работы с ресурсом:

```
GET /version HTTP/1.1
Host: 127.0.0.1
Accept: application/xml
HTTP/1.1 200 OK
Date: Mon, 23 May 2005 22:38:34 GMT
Content-Type: application/xml

<?xml version="1.0" encoding="utf-8"?>
```

```
<version>
    <major>2</major>
    <minor>37</minor>
</version>
```

Изменения спецификации

Версия 1.0

- Имена XML-представлении приведены к единому формату — нижний регистр символов, символ «-» в качестве разделителя слов.
- Параметры камеры (разрешение, цветность и т.д.) не выдаются в представлении камеры.
- Определены значения по умолчанию для параметров запроса изображений (разрешение, качество и т. д.).
- Версия протокола 0.0 не поддерживается.

Версия 1.1

- Добавлена поддержка объекта «микрофон».
- Исправлена ошибка описания имен тегов вида `<xxx-uri>` (в предыдущей версии были указаны как `<xxx-url>`).

Версия 1.2

- Добавлено поле `<uri>` в объект Camera.
- Добавлено поле `<uri>` в объект Microphone.
- Добавлена поддержка объекта OSD (on-screen display).
- Добавлено поле `<osd-uri>` в объект Camera.

Версия 1.3

- Добавлен цвет текста (поле `<font-color>`) в объект OSD.

Версия 1.4

- Добавлены поля `<width>` и `<height>` в объект Camera.
- Добавлен параметр `keep_aspect_ratio` в объект «Изображение».

Версия 1.5

- Добавлено поле `<draw-background>` в объект OSD.