

Остановка видеопотока со стороны WCS-сервера

Описание

Любой поток, опубликованный или захваченный на WCS, может быть остановлен на стороне сервера.

Для остановки публикации или воспроизведения потока со стороны WCS-сервера доступны два способа:

1. Возврат кода состояния HTTP `403 FORBIDDEN` на keep-alive запрос для потока
2. REST-запрос к WCS-серверу

Keep-Alive видеопотока

Keep-alive сообщения могут быть использованы для завершения потоков со стороны WCS-сервера.

Для этого бэкенд сервер, на котором сконфигурирован REST Hooks, должен вернуть код состояния HTTP `403 FORBIDDEN` на keep-alive запрос для потока.

Таким образом могут быть остановлены публикация или воспроизведение потока, или и то, и другое. Чтобы различить опубликованные и воспроизводимые потоки, скрипт бэкенд сервера должен обрабатывать `published` параметр метода `StreamKeepAliveEvent`

Настройка REST hook

Сконфигурируйте web-сервер с [REST Hooks](#). Он должен обрабатывать keep-alive запросы от WCS-сервера, используя, например, PHP-скрипт, и определять `restClientConfig` для метода `StreamKeepAliveEvent`

```
"StreamKeepAliveEvent" : {
  "clientExclude" : "",
  "restExclude" : "sdp",
  "restOnError" : "FAIL",
  "restPolicy" : "NOTIFY",
  "restOverwrite" : ""
}
```

Серверная часть

Чтобы включить отправку keep-alive сообщений для потоков, необходимо:

1. Включить настройку keep-alive в `flashphoner.properties`

```
keep_alive_streaming_sessions_enabled=true
```

2. Определить keep-alive интервал и приложение, которое будет принимать ответы на keep-alive REST-запросы

```
streaming_sessions_keep_alive_interval=10000  
streaming_sessions_keep_alive_app_keys=defaultApp
```

Может быть использовано другое приложение. Используйте [интерфейс командной строки](#), команду `show apps`, чтобы просмотреть список приложений и их ключи

3. Добавить к этому приложению REST-метод `StreamKeepAliveEvent`, используя следующую команду

```
add app-rest-method defaultApp StreamKeepAliveEvent
```

4. Подключить REST hook к приложению из командной строки

```
update app defaultApp http://my-web-server.com/MyAPI
```

Здесь:

5. `my-web-server.com` - это адрес бэкенд сервера,
6. `MyAPI` - это путь, по которому расположен REST hook.

REST-вызовы для остановки потока

Для остановки потока используется REST запрос `/stream/terminate`

REST запрос должен быть HTTP/HTTPS POST запросом в таком виде:

- HTTP: `http://streaming.flashphoner.com:8081/rest-api/stream/terminate`
- HTTPS: `https://streaming.flashphoner.com:8444/rest-api/stream/terminate`

Здесь:

- `streaming.flashphoner.com` - адрес WCS-сервера
- `8081` - стандартный REST / HTTP порт WCS-сервера
- `8444` - стандартный HTTPS порт

- `rest-api` - обязательная часть URL
- `/stream/terminate` - используемый REST-метод

REST-методы и статусы ответа

REST запрос	Тело запроса	Статусы ответа
<code>/stream/terminate`</code>	<pre>{ "mediaSessionId": "41c3f621-a847-4639" }</pre>	200 Поток будет остановлен 404 Поток не найден

Параметры

Параметр	Описание	Пример
<code>mediaSessionId</code>	Идентификатор медиа сессии	<code>`41c3f621-a847-4639`</code>
<code>name</code>	Имя поток	<code>`streamName`</code>
<code>published</code>	Если <code>`true`</code> , то поток является опубликованным; если <code>`false`</code> , то поток воспроизводится	<code>`true`</code>
<code>status</code>	Текущий статус потока	<code>`PUBLISHING`</code>

Фильтрация потоков по параметрам

Параметры запроса `/stream/terminate` рассматриваются как фильтры, будут остановлены все потоки, соответствующие этим фильтрам. Например, можно остановить всех подписчиков для всех опубликованных потоков

```
{
  "published": false
}
```

или всех подписчиков для определенного потока

```
{
  "name": "streamName",
```

```
}
  "published": false
}
```

Можно остановить публикацию по статусу

```
{
  "name": "streamName",
  "status": "PUBLISHING"
}
```

Также можно остановить все потоки в определенных медиасессиях

```
{
  "mediaSessionIds": [
    "41c3f621-a847-4639",
    "554916e0-931c-2479"
  ]
}
```

или в одной медиасессии

```
{
  "mediaSessionId": "41c3f621-a847-4639"
}
```

Отправка REST-запроса к WCS-серверу

Для отправки REST-запроса к WCS-серверу необходимо использовать REST-клиент, например расширение [Advanced REST Console](#) для браузера Chrome.

Возобновление потока

После остановки потока со стороны сервера публикация и воспроизведение потока могут быть возобновлены. Это будет новый поток.

Например, в случае WebRTC потока, опубликованного с использованием клиента `client2/examples/min/streaming.html` и воспроизводимого с использованием клиента `client2/examples/demo/streaming/player/player.html`:

1. Поток опубликован

WCS Server URL:


Stream name:

Name: streamName Status: PUBLISHING



2. Поток воспроизводится

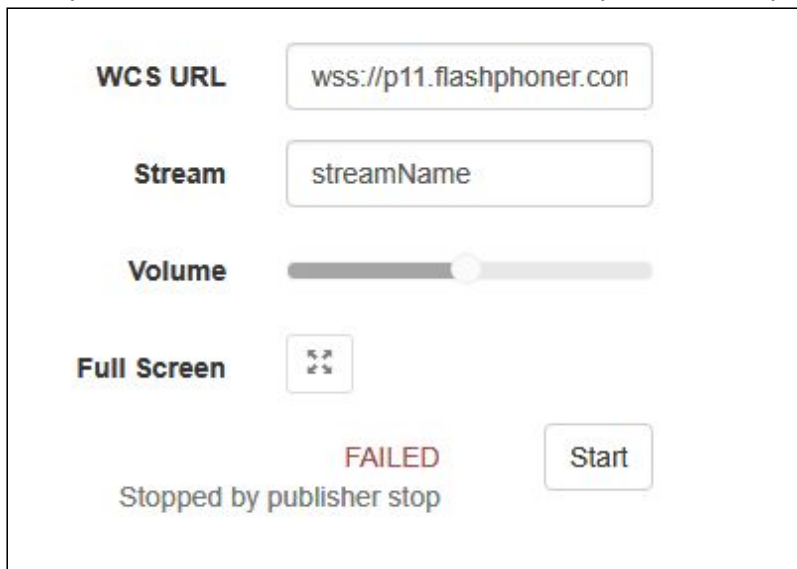
Player



WCS URL

Stream

3. Воспроизведение потока остановлено со стороны WCS-сервера



Публикация потока будет возобновлена после нажатия кнопки `publish`.
Воспроизведение потока будет возобновлено после нажатия кнопки `Start`.

Известные проблемы

1. Сообщения `Can't find mediasession` в логе клиента



Симптомы

В случае остановки со стороны сервера RTMFP-потока, опубликованного с использованием клиента `client2/examples/demo/streaming/flash_client/chat.html`, публикация потока останавливается, однако сессия потока не уничтожается. В [логе клиента](#) появляются сообщения `Can't find mediasession`



Решение

Остановить сессию после завершения потока