Вставка одного потока в другой

Описание

В сборке 5.2.841 добавлена возможность вставки одного опубликованного на сервере потока в другой. Эту функцию можно использовать, например, для добавления рекламы в поток. При этом содержимое одного потока полностью заменяется другим, либо до окончания публикации второго потока, либо до прекращения вставки.

Поддерживаемые кодеки

Видео:

- H264
- VP8

Аудио:

- Opus
- AAC
- G711

Ограничения

1. Оба потока, к которым применяется вставка, должны быть закодированы одинаковыми аудио и видео кодеками.
2. Для аудио, должна быть одинаковая частота дискретизации и одинаковое количество каналов.
3. Вставка не применяется к потокам звонков. Для звонков используются собственные технологии вставки аудио и видео.
4. В один поток может быть вставлен только один поток одновременно, но один и тот же поток может быть вставлен в несколько потоков.
5. Циклическая вставка не поддерживается. Невозможно вставить `stream1` в `stream2`, а затем `stream2` в `stream1` без остановки предыдущей вставки.

Реализация вставки в сборках до 5.2.1618

REST API

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

- HTTP: `http://test.flashphoner.com:8081/rest-api/stream/inject/startup`
- HTTPS: `https://test.flashphoner.com:8444/rest-api/stream/inject/startup`

Здесь:

- `test.flashphoner.com` - адрес WCS-сервера
- `8081` - стандартный REST / HTTP порт WCS-сервера
- `8444` - стандартный HTTPS порт
- `rest-api` - обязательная часть URL
- `/stream/inject/startup` - используемый REST-метод

REST методы и ответы

/stream/inject/startup

Вставить поток stream2 в stream1

REQUEST EXAMPLE

```
POST /rest-api/stream/inject/startup HTTP/1.1
Host: localhost:8081
Content-Type: application/json

{
    "localStreamName": "stream1",
    "remoteStreamName": "stream2"
}
```

RESPONSE EXAMPLE

```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
Content-Type: application/json
```

RETURN CODES

Code	Reason
200	OK
400	Bad request
404	Not found
409	Conflict

Code	Reason
500	Internal error

/stream/inject/find_all

Найти все вставки на сервере

REQUEST EXAMPLE

```
POST /rest-api/stream/inject/find_all HTTP/1.1
Host: localhost:8081
Content-Type: application/json
```

RESPONSE EXAMPLE

```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
Content-Type: application/json

[
    {
        "localStreamName": "stream1",
        "remoteStreamName": "stream2"
    }
]
```

RETURN CODES

Code	Reason
200	OK
404	Not found

`stream/inject/terminate

Остановить вставку в поток stream1

REQUEST EXAMPLE

```
POST /rest-api/stream/inject/terminate HTTP/1.1
Host: localhost:8081
Content-Type: application/json

{
    "localStreamName": "stream1"
}
```

RESPONSE EXAMPLE

```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
Content-Type: application/json
```

RETURN CODES

Code	Reason
200	OK
400	Bad request
404	Not found
500	Internal error

Parameters

Name	Description	Example
localStreamName	Имя потока, в который производится вставка	`stream1`
remoteStreamName	Имя потока, который бу дет вставлен	`stream2`

Вставка VOD потока из файла

В сборке 5.2.1535 добавлена возможность вставить VOD поток непосредственно из файла при отправке запроса `/stream/inject/startup`:

```
{
    "localStreamName":"host",
    "remoteStreamName":"vod-live://advertising.mp4"
}
```

При этом вставляемый файл начинает проигрываться без пауз, с первого ключевого кадра. Если этот же файл в ставить в другой поток, в том потоке файл также начнет проигрываться с начала.

Эта возможность полезна, например, при вставке рекламных роликов в поток, который смотрят зрители.

Настройка

В сборке 5.2.1235 добавлена настройка, которая определяет, в течение какого времени в миллисекундах необходимо ждать ключевого кадра во вставляемом

потоке

```
inject_wait_keyframe_ms=1000
```

По умолчанию, интервал составляет 1000 миллисекунд. Если ключевой кадр во вставляемом потоке за это время не был получен, сервер начинает генерировать черный фон (по умолчанию), либо кадр с изображением из файла, заданного в настройке `custom_watermark_filename` . Это поведение можно отключить настройкой

```
inject_wait_keyframe_ms=-1
```

В этом случае будет продолжаться проигрывание потока, в который производится вставка, до момента получения ключевого кадра во вставляемом потоке.

Реализация вставки в сборке 5.2.1618 и новее

Настройка

В сборке 5.2.1618 добавлена новая реализация вставки одного потока в другой, позволяющая выбрать, какую именно составляющую заменить: аудио, видео или обе. Эта возможность включается настройкой

```
use_new_injector=true
```

REST API

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

- HTTP: `http://test.flashphoner.com:8081/rest-api/stream/inject2/startup`
- HTTPS: `https://test.flashphoner.com:8444/rest-api/stream/inject2/startup`

Здесь:

- `test.flashphoner.com` - адрес WCS-сервера
- `8081` - стандартный REST / HTTP порт WCS-сервера
- `8444` - стандартный HTTPS порт
- `rest-api` - обязательная часть URL
- `/stream/inject2/startup` - используемый REST-метод

REST методы и ответы

/stream/inject2/startup

Вставить поток stream2 в поток stream1

REQUEST EXAMPLE

```
POST /rest-api/stream/inject2/startup HTTP/1.1
Host: localhost:8081
Content-Type: application/json

{
    "localStreamName": "stream1",
    "remoteStreamName": "stream2",
    "video": true,
    "audio": true,
    "muteIfAbsent": true
}
```

RESPONSE EXAMPLE

```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
Content-Type: application/json
```

RETURN CODES

Code	Reason
200	OK
400	Bad request
404	Not found
409	Conflict
500	Internal error

/stream/inject2/find_all

Найти все вставки на сервере

REQUEST EXAMPLE

```
POST /rest-api/stream/inject2/find_all HTTP/1.1
Host: localhost:8081
Content-Type: application/json
```

RESPONSE EXAMPLE

```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
Content-Type: application/json

[
```

```
  {
    "streamName": "test",
    "videoInjectorInfo": {
      "targetStreamName": "test2",
      "rootStreamName": "test2",
      "startTime": 1683344295099
    },
    "audioInjectorInfo": {
      "targetStreamName": "test2",
      "rootStreamName": "test2",
      "startTime": 1683344295056
    }
  }
]
```

RETURN CODES

Code	Reason
200	OK
404	Not found

/stream/inject2/terminate

Остановить вставку в поток stream1

REQUEST EXAMPLE

```
POST /rest-api/stream/inject2/terminate HTTP/1.1
Host: localhost:8081
Content-Type: application/json

{
    "localStreamName": "stream1",
    "video": true,
    "audio": true
}
```

RESPONSE EXAMPLE

```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
Content-Type: application/json
```

RETURN CODES

Code	Reason
200	OK
400	Bad request

Code	Reason
404	Not found
500	Internal error

Parameters

Name	Description	Example
localStreamName	Имя потока, в который производится вставка	`stream1`
remoteStreamName	Имя потока, который будет вставлен	`stream2`
video	Заменять видеосоставляющую при вставке потока	`true`
audio	Заменять аудиосоставляющую при вставке потока	`true`
muteIfAbsent	Заменять составляющую, которой нет в исходном потоке, на темноту или тишину	`true`
videoInjectorInfo	Информация о видео из вставленного потока	{ "targetStreamName": "stream2", "rootStreamName": "stream2", "startTime": 1683344295099 }

Name	Description	Example

audioInjectorInfo	Информация об аудио из вставленного потока	{
		"targetStreamName": "stream2",
		"rootStreamName": "stream2",
		"startTime": 1683344295056
		}

Вставка VOD потока из файла

В сборке 5.2.1719 добавлена возможность вставить VOD поток непосредственно из файла при отправке запроса `/stream/inject2/startup`:

```
{
    "localStreamName":"host",
    "remoteStreamName":"vod-live://advertising.mp4",
    "video":true,
    "audio":true
}
```

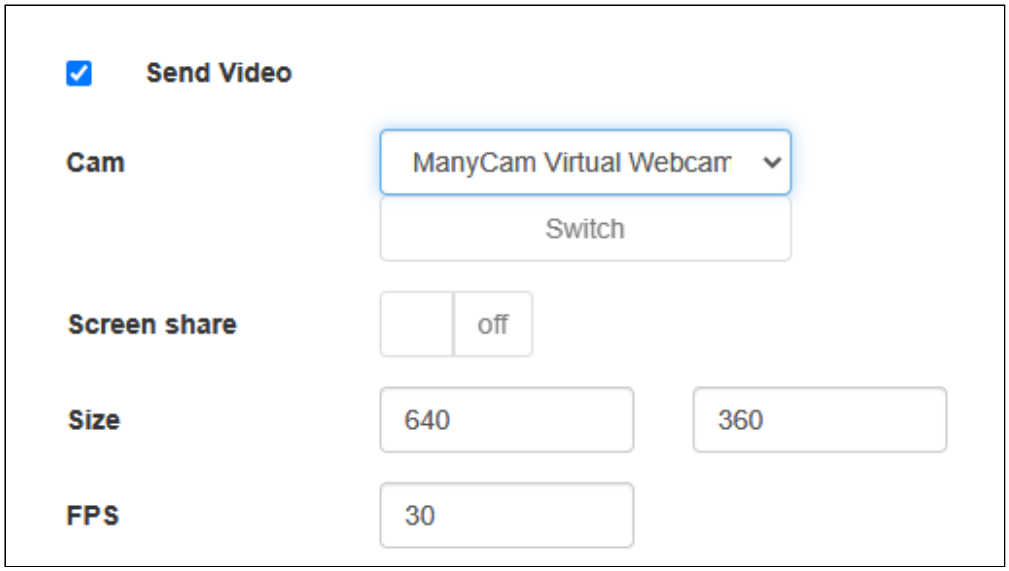
При этом вставляемый файл начинает проигрываться без пауз, с первого ключевого кадра. Если этот же файл вставить в другой поток, в том потоке файл также начнет проигрываться с начала

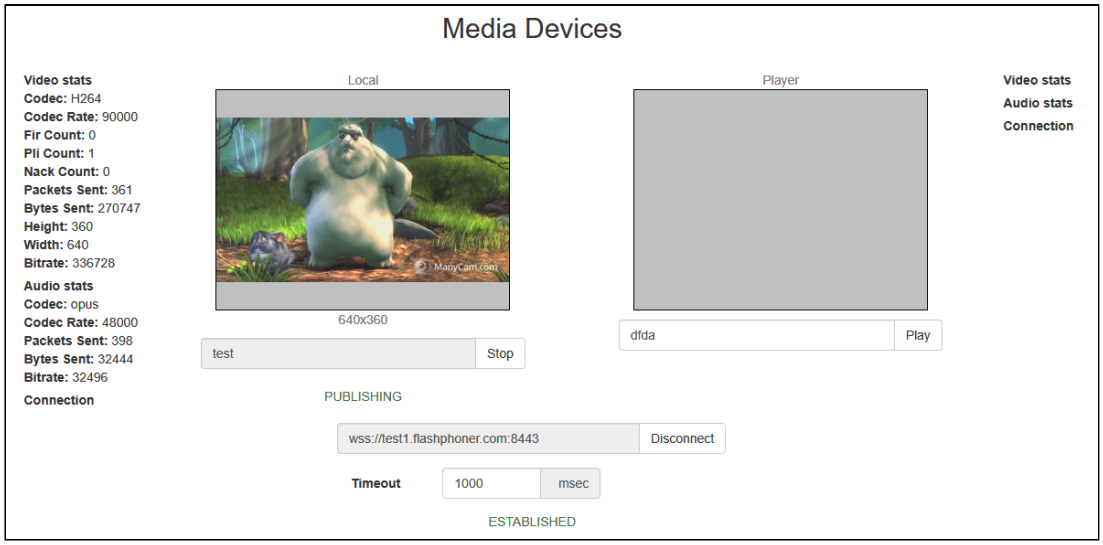
Эта возможность полезна, например, при вставке рекламных роликов в поток, который смотрят зрители.

Краткое руководство по тестированию

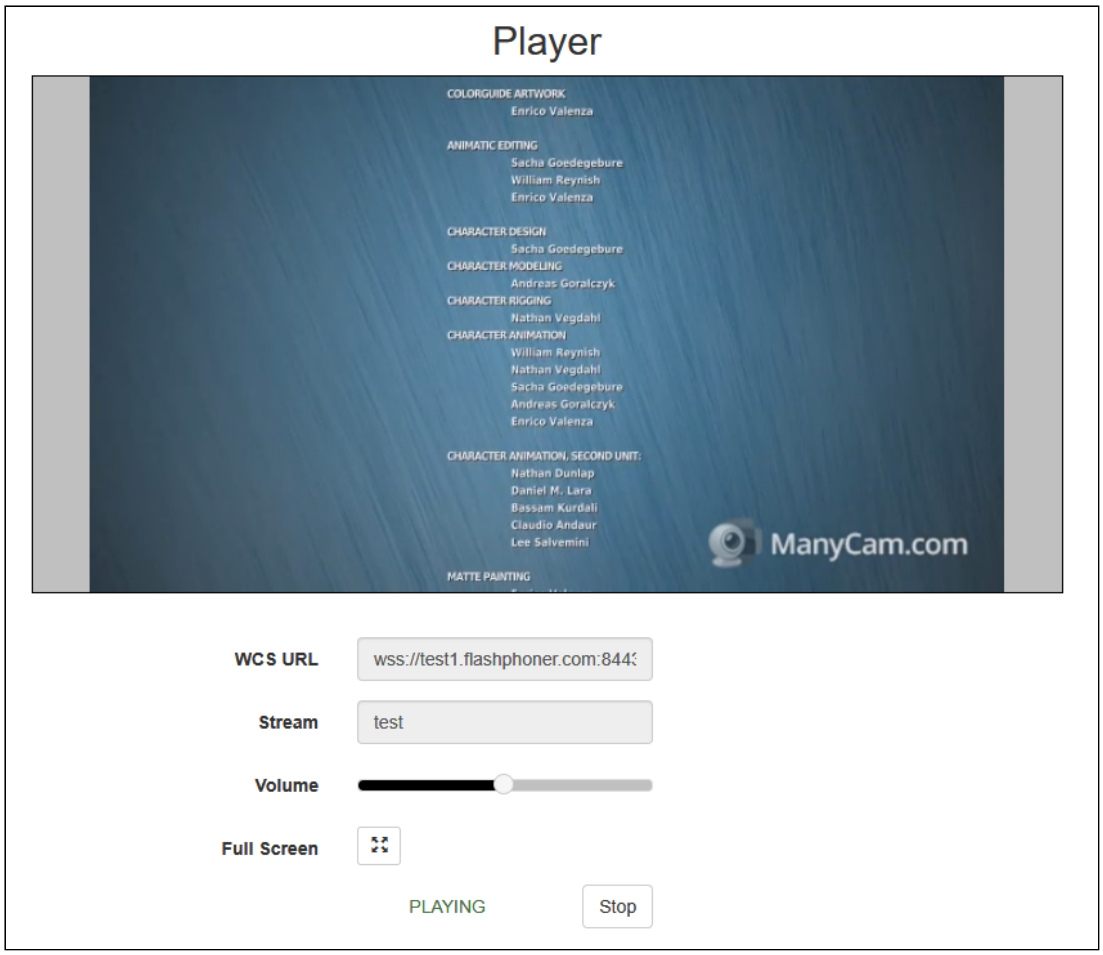
1. Для тестирования используем:
    - WCS-сервер;
    - Веб-приложение Media Devices для публикации потоков;
    - Две веб-камеры, либо два различных ПК для публикации потоков;
    - Веб-приложение Player для воспроизведения потока;
    - браузер Chrome и REST-клиент для отправки запросов на сервер

2. Откройте приложение Media Devices, опубликуйте поток `test` разрешением 640x360

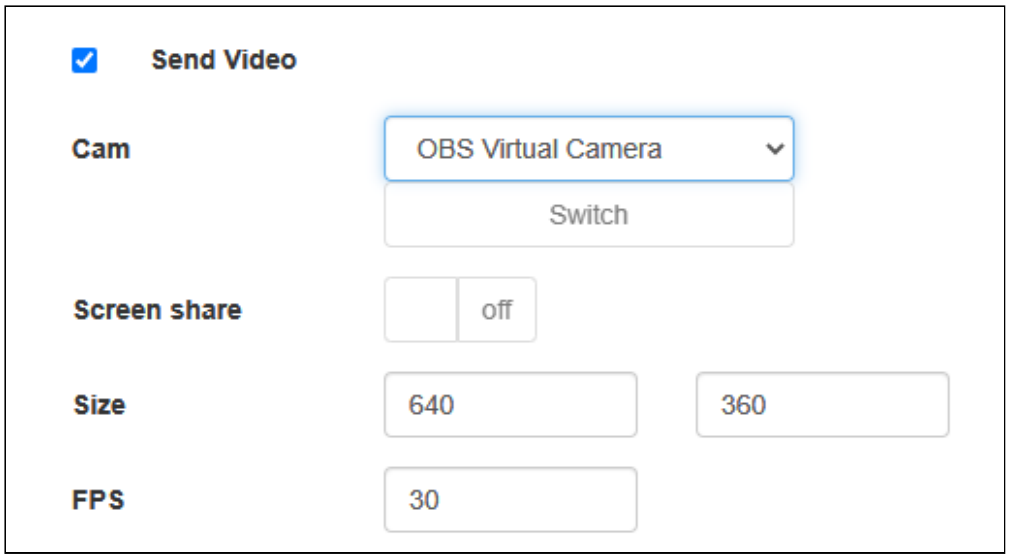


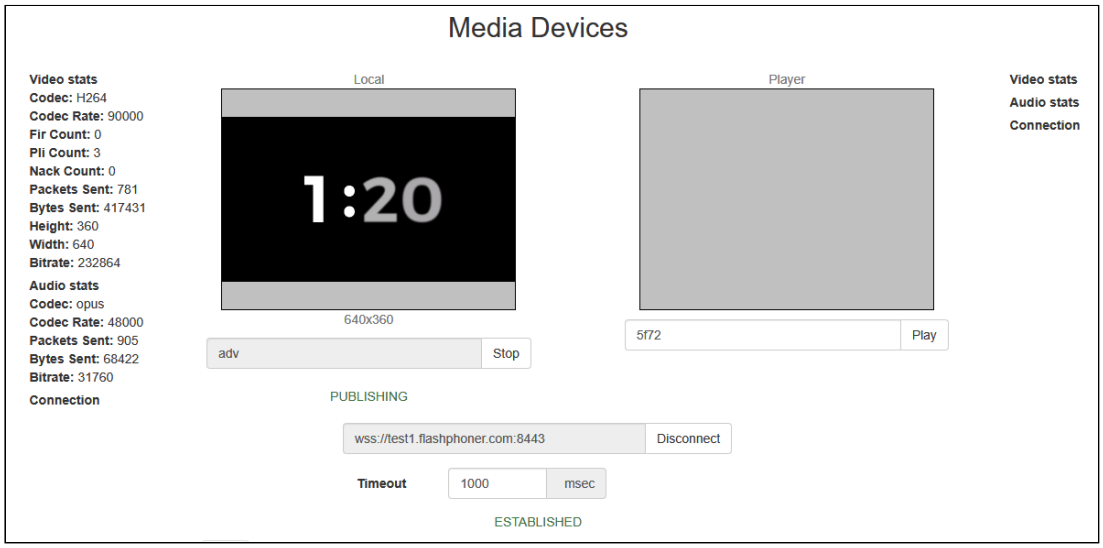


3. Проиграйте поток `test` в примере Player

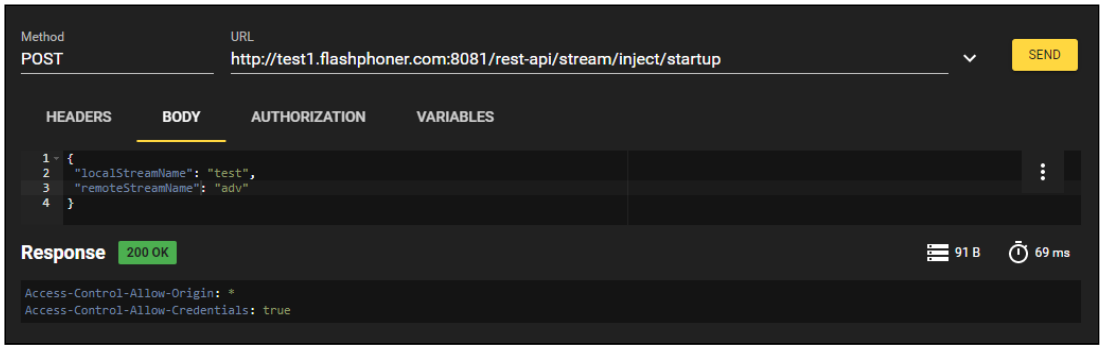


4. Опубликуйте поток `adv` в примере Media Devices, используя другую вкладку браузера, другую камеру или другой ПК

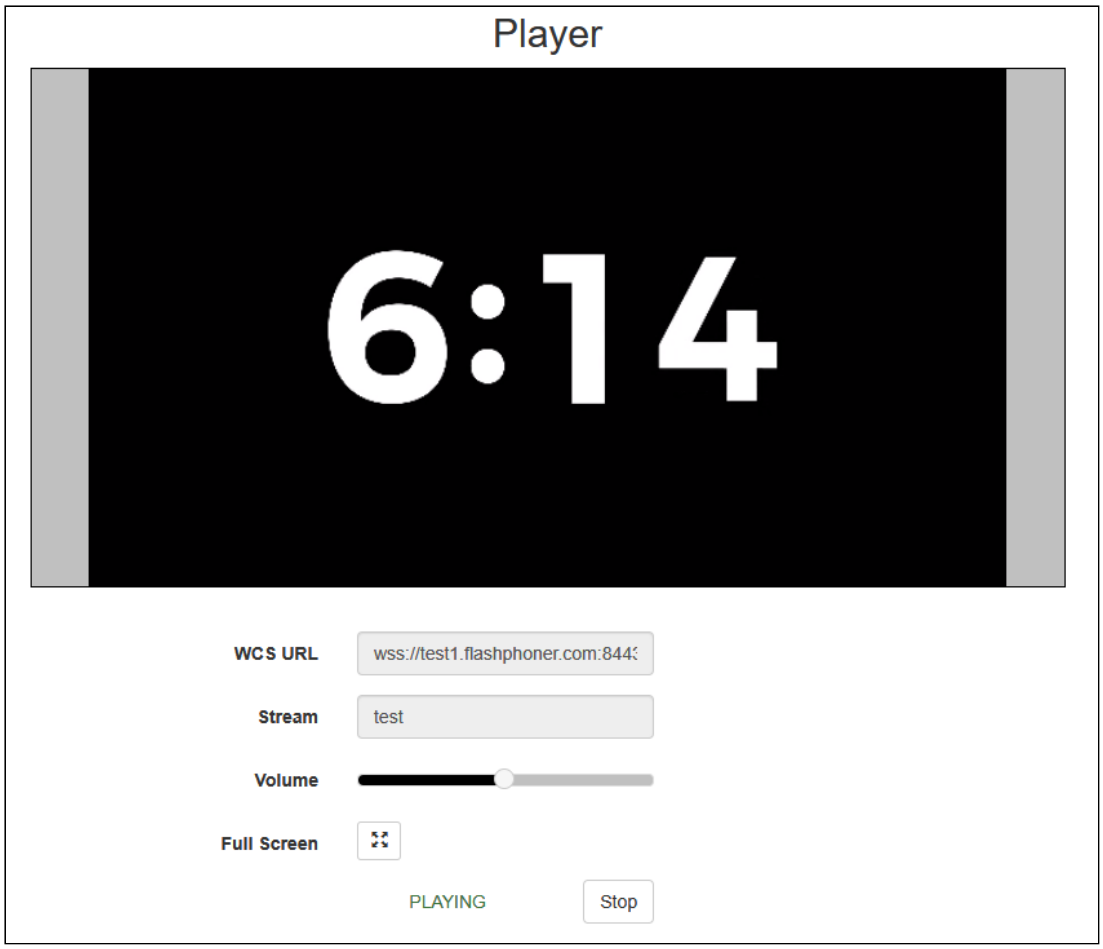




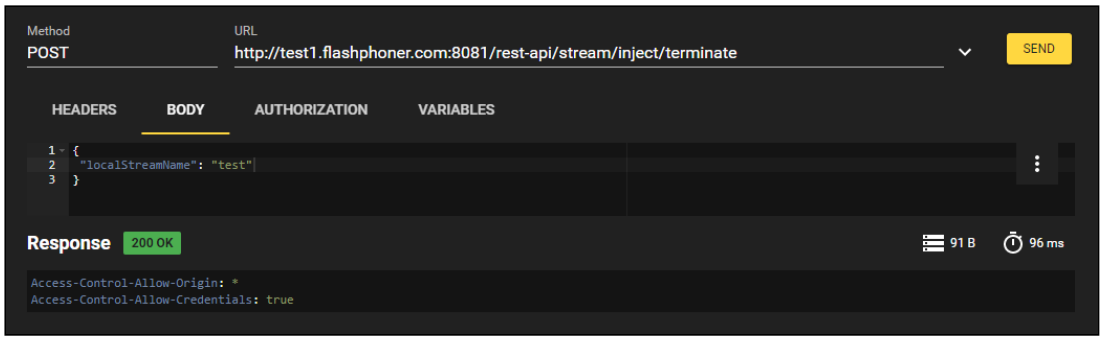
5. Откройте REST-клиент, отправьте запрос `/stream/inject/startup`



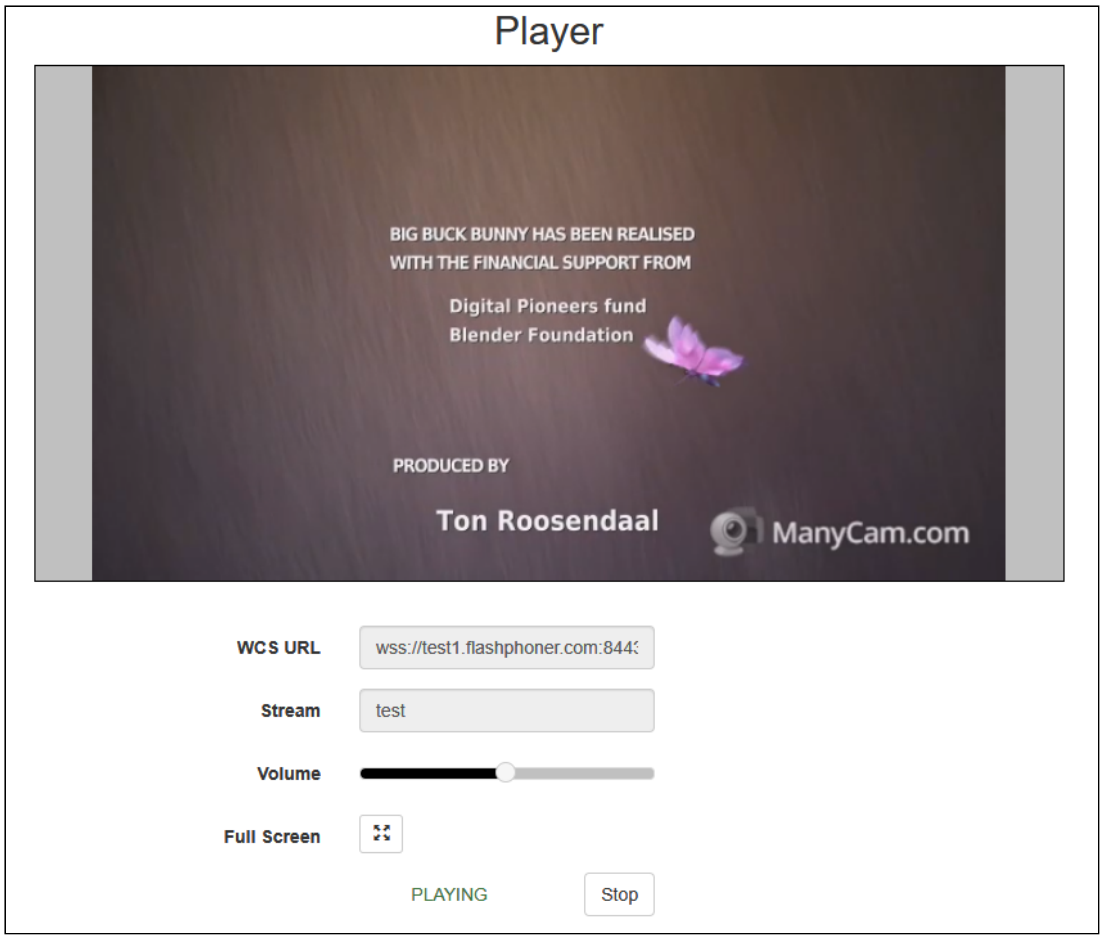
6. В потоке test воспроизводится содержимое потока `adv`



7. Отправьте запрос `/stream/inject/terminate`



8. В потоке `test` вновь играет оригинальное содержимое



Известные проблемы

1. По окончании вставки одного RTMP потока в другой может теряться синхронизация между аудио и видео а оригинальном потоке

Симптомы

При вставке RTMP потока в другой RTMP поток, по окончании вставки оригинальный поток играет с рассинхронизацией аудио и видео

Решение

Включить буферизацию входящих RTMP потоков

```
rtmp_in_buffer_enabled=true
```