

Нагрузочное тестирование микшера

Описание

Если сценарий использования сервера предусматривает микширование потоков, возникает необходимость нагрузочного тестирования микшера перед вводом сервера в эксплуатацию. Тестирование проводится следующим образом:

1. На сервере публикуется необходимое количество потоков (минимум по одному потоку на микшер)
2. Создается указанное количество аудиомикшеров, на вход которым подается заданное количество потоков. Один поток может быть подан на вход только одного микшера.
3. Микшеры работают в течение заданного времени, после чего уничтожаются и создаются заново, до окончания тестирования.

В процессе тестирования можно наблюдать за поведением сервера при помощи средств [мониторинга](#).

Для управления тестированием предназначены специальные REST API запросы.

REST API

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

- HTTP: `http://streaming.flashphoner.com:8081/rest-api/mixer/test/start`
- HTTPS: `https://streaming.flashphoner.com:8444/rest-api/mixer/test/start`

Здесь:

- `streaming.flashphoner.com` - адрес WCS-сервера
- `8081` - стандартный REST / HTTP порт WCS-сервера
- `8444` - стандартный HTTPS порт
- `rest-api` - обязательный префикс
- `mixer/test/start` - используемый REST-вызов

REST методы

`/mixer/test/start`

Запустить тест

REQUEST EXAMPLE

```
POST /mixer/test/start HTTP/1.1
Host: localhost:8081
Content-Type: application/json

{
  "feedingStreams": [
    "s1",
    "s2",
    "s3",
    "s4"
  ],
  "mixerCount": 2,
  "streamsInMixer": 2,
  "intervalInSeconds": 60
}
```

RESPONSE EXAMPLE

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

RETURN CODES

Code	Reason
200	OK
500	Internal error

/mixer/test/stop

Остановить тест

REQUEST EXAMPLE

```
POST /mixer/test/stop 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
```

RETURN CODES

Code	Reason
200	OK
404	Mixer not found
500	Internal error

`/mixer/test/get_start_example`

Возвращает пример JSON объекта для вызова `/mixer/test/start`

REQUEST EXAMPLE

```
POST /mixer/test/get_start_example HTTP/1.1
Host: localhost:8081
Content-Type: application/json

{
  "feedingStreams": [
    "stream1",
    "stream2",
    "stream3"
  ],
  "mixerCount": 3,
  "streamsInMixer": 1,
  "intervalInSeconds": 60
}
```

RESPONSE EXAMPLE

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

RETURN CODES

Code	Reason
200	OK
500	Internal error

Параметры

Parameter name	Description	Example
----------------	-------------	---------

Parameter name	Description	Example
feedingStreams	Список опубликованных на сервере потоков, которые будут участвовать в тестировании	<pre>["s1", "s2", "s3", "s4"]</pre>
mixerCount	Количество создаваемых микшеров	2
streamsInMixer	Количество потоков, направляемых в каждый микшер	2
intervalInSeconds	Интервал в секундах, после которого микшеры уничтожаются и создаются снова	60

Настройка

Для тестирования производительности микшера необходимо отключить асинхронное завершение медиасессий (которое работает по умолчанию)

```
handler_async_disconnect=false
```

и перезапустить WCS.

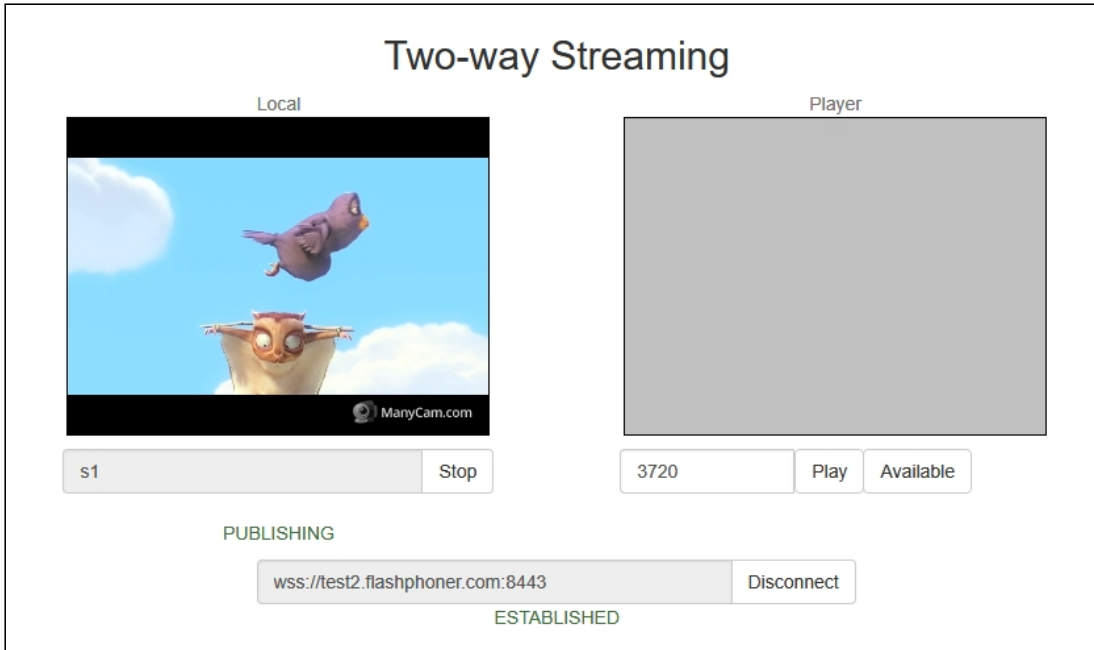
По окончании тестирования данную настройку необходимо убрать.

Тестирование

1. Для теста используем:

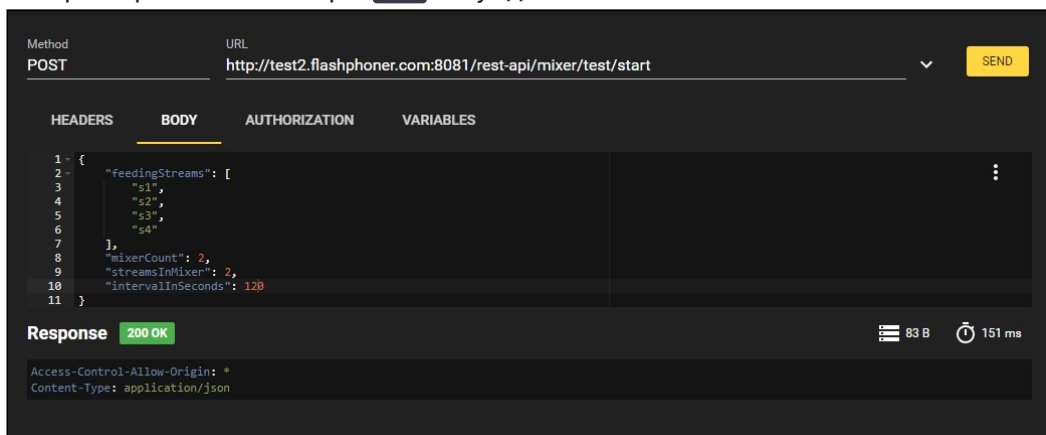
- WCS сервер
- браузер Chrome и [REST-клиент](#) для отправки запросов
- веб-приложение `Two Way Streaming` для публикации потоков

2. Опубликуйте потоки с именами `s1`, `s2`, `s3`, `s4`

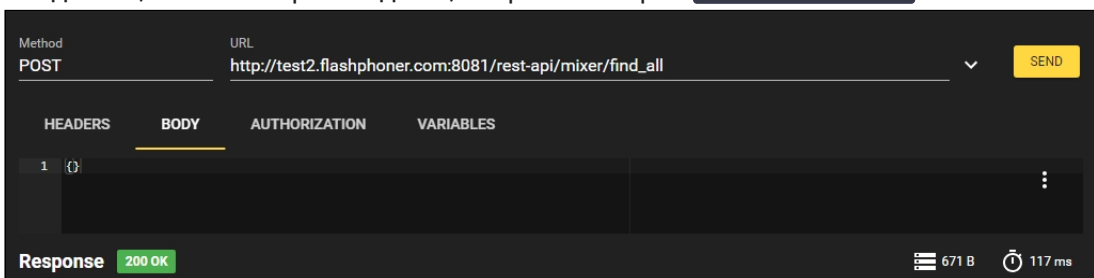


3. Откройте REST-клиент. Отправьте запрос `/mixer/test/start`, указав в параметрах:

- список опубликованных потоков: `s1, s2, s3, s4`
- количество микшеров: `2`
- количество потоков на микшер: `2`
- интервал работы микшера: `120` секунд



4. Убедитесь, что микшеры созданы, отправив запрос `/mixer/find_all`



Должно быть найдено два микшера `mixer0` и `mixer1`

```
Access-Control-Allow-Origin: *
Content-Type: application/json
Content-Length: 567

  2 | L
  3 | {
  4 |   "localMediaSessionId": "70783957-e03f-46cc-975e-a9fe174950a7",
  5 |   "localStreamName": "mixer1",
  6 |   "uri": "mixer://mixer1",
  7 |   "status": "PROCESSED_LOCAL",
  8 |   "hasAudio": true,
  9 |   "hasVideo": true,
 10 |   "record": false,
 11 |   "mediaSessions": [
 12 |     "d619b1d0-a5b2-11e8-8348-ebb0edd6d2e8",
 13 |     "ca05f570-a5b2-11e8-8508-b1c91d8ebd18"
 14 |   ],
 15 | },
 16 | {
 17 |   "localMediaSessionId": "7be960fc-08fe-444f-86ee-c3ceb1d87ac5",
 18 |   "localStreamName": "mixer0",
 19 |   "uri": "mixer://mixer0",
 20 |   "status": "PROCESSED_LOCAL",
 21 |   "hasAudio": true,
 22 |   "hasVideo": true,
 23 |   "record": false,
 24 |   "mediaSessions": [
 25 |     "af49ef20-a5b2-11e8-bf17-ab9c5e415872",
 26 |     "bba6c270-a5b2-11e8-8b25-f7438838b4d6"
```

5. Во время теста можно наблюдать за потреблением ресурсов сервера при помощи [Java Mission Control](#), страниц [информации о нагрузке и ресурсах](#) и [информации об ошибках](#), а также [серверных логов](#).

6. Остановите тест, отправив запрос `/mixer/test/stop`

```
Method: POST
URL: http://test2.flashphoner.com:8081/rest-api/mixer/test/stop
SEND

HEADERS
BODY
AUTHORIZATION
VARIABLES

1 {}

Response 200 OK 83 B 61 ms
Access-Control-Allow-Origin: *
Content-Type: application/json
```

Рекомендации по настройке

1. Если во время тестирования была выявлена большая загрузка CPU, следуйте [рекомендациям по тонкой настройке сервера](#).
2. Если во время тестирования была выявлена утечка ресурсов, направьте описание теста и [собранный отчет](#) на support@flashphoner.com