

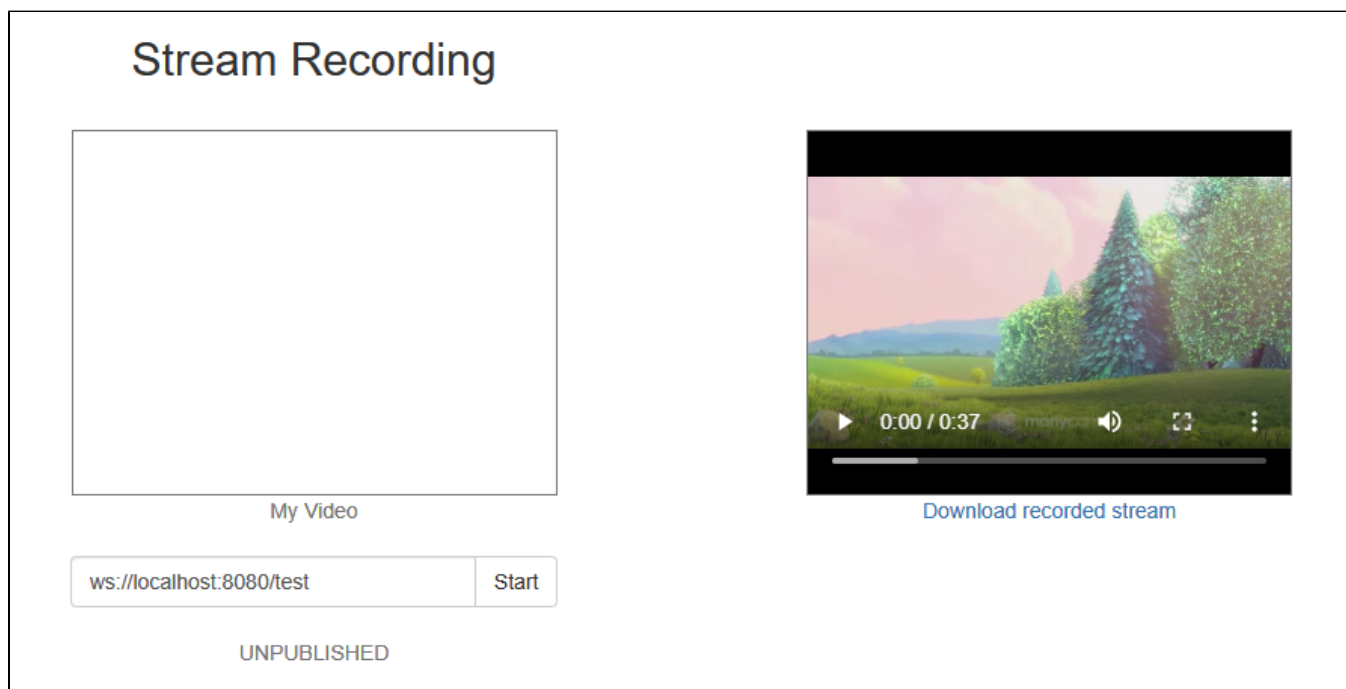
Stream Recording

- [Пример стримера с записью видеопотока](#)
- [Код примера](#)
- [Работа с кодом примера](#)

Пример стримера с записью видеопотока

Данный стример может использоваться с Web Call Server для публикации и записи WebRTC потока

На скриншоте ниже представлен пример с завершенной публикацией потока.



Во время публикации в 'My Video' воспроизводится видео с камеры.

По завершении публикации запись видеопотока становится доступной для скачивания и воспроизведения в плеере справа.

Код примера

Код данного примера находится на WCS-сервере по следующему пути:

```
/usr/local/FlashphonerWebCallServer/client2/examples/demo/streaming/stream_recording
```

recording.css - файл стилей

recording.html - страница стримера

recording.js - скрипт, обеспечивающий работу стримера

Тестировать данный пример можно по следующему адресу:

```
https://host:8888/client2/examples/demo/streaming/stream_recording/recording.html
```

Здесь host - адрес WCS-сервера.

Работа с кодом примера

Для разбора кода возьмем версию файла recording.js с хешем ecbadc3, которая находится [здесь](#) и доступна для скачивания в соответствующей сборке [2.0.212](#).

1. Инициализация API.

Flashphoner.init() [code](#)

```
Flashphoner.init();
```

2. Подключение к серверу.

Flashphoner.createSession() [code](#)

```
Flashphoner.createSession({urlServer: url}).on(SESSION_STATUS.ESTABLISHED, function(session){
    ...
}).on(SESSION_STATUS.DISCONNECTED, function(){
    ...
}).on(SESSION_STATUS.FAILED, function(){
    ...
});
```

3. Получение от сервера события, подтверждающего успешное соединение.

ConnectionStatusEvent ESTABLISHED [code](#)

```
Flashphoner.createSession({urlServer: url}).on(SESSION_STATUS.ESTABLISHED, function(session){
    setStatus(session.status());
    //session connected, start playback
    publishStream(session);
}).on(SESSION_STATUS.DISCONNECTED, function(){
    ...
}).on(SESSION_STATUS.FAILED, function(){
    ...
});
```

4. Публикация видеопотока.

session.createStream(), publish() [code](#)

При создании передаются параметры

- streamName - имя видеопотока
- localVideo - div-элемент, в котором будет отображаться видео с камеры
- record: true - для того, чтобы была сделана запись потока

```
session.createStream({
    name: streamName,
    display: localVideo,
    record: true
    ...
}).publish();
```

5. Получение от сервера события, подтверждающего успешную публикацию.

StreamStatusEvent PUBLISHING [code](#)

```
session.createStream({
    ...
}).on(STREAM_STATUS.PUBLISHING, function(stream) {
    setStatus(stream.status());
    onStart(stream);
}).on(STREAM_STATUS.UNPUBLISHED, function(stream) {
    ...
}).on(STREAM_STATUS.FAILED, function(stream) {
    ...
}).publish();
```

6. Остановка публикации видеопотока.

stream.stop() [code](#)

```
function onStarted(stream) {
    $("#publishBtn").text("Stop").off('click').click(function(){
        $(this).prop('disabled', true);
        stream.stop();
    }).prop('disabled', false);
}
```

7. Получение от сервера события, подтверждающего успешную остановку публикации.

StreamStatusEvent UNPUBLISHED [code](#)

При обработке события вызывается функция формирования ссылки для скачивания файла записи showDownloadLink(), в функцию передается имя записанного файла, получаемое при помощи метода stream.getRecordInfo()

```
session.createStream({
    ...
}).on(STREAM_STATUS.PUBLISHING, function(stream) {
    ...
}).on(STREAM_STATUS.UNPUBLISHED, function(stream) {
    setStatus(stream.status());
    showDownloadLink(stream.getRecordInfo());
    onStopped();
}).on(STREAM_STATUS.FAILED, function(stream) {
    ...
}).publish();
```

8. Формирование ссылки для скачивания.

[code](#)

```
function showDownloadLink(name) {
    if (name) {
        ...
        var link = window.location.protocol + "://" + window.location.host + '/client/records/' + name;
        $("#link").attr("href", link);
        $("#recVideo").attr("src", link).attr("controls", true);
        $("#downloadDiv").show();
    }
}
```

Записи потоков сохраняются в директорию WCS_HOME/client/records.

Соответственно, при формировании ссылки

- window.location.protocol - используемый протокол (HTTP или HTTPS)
- window.location.host - <адрес WCS-сервера>:<порт>
- name - имя файла с записью