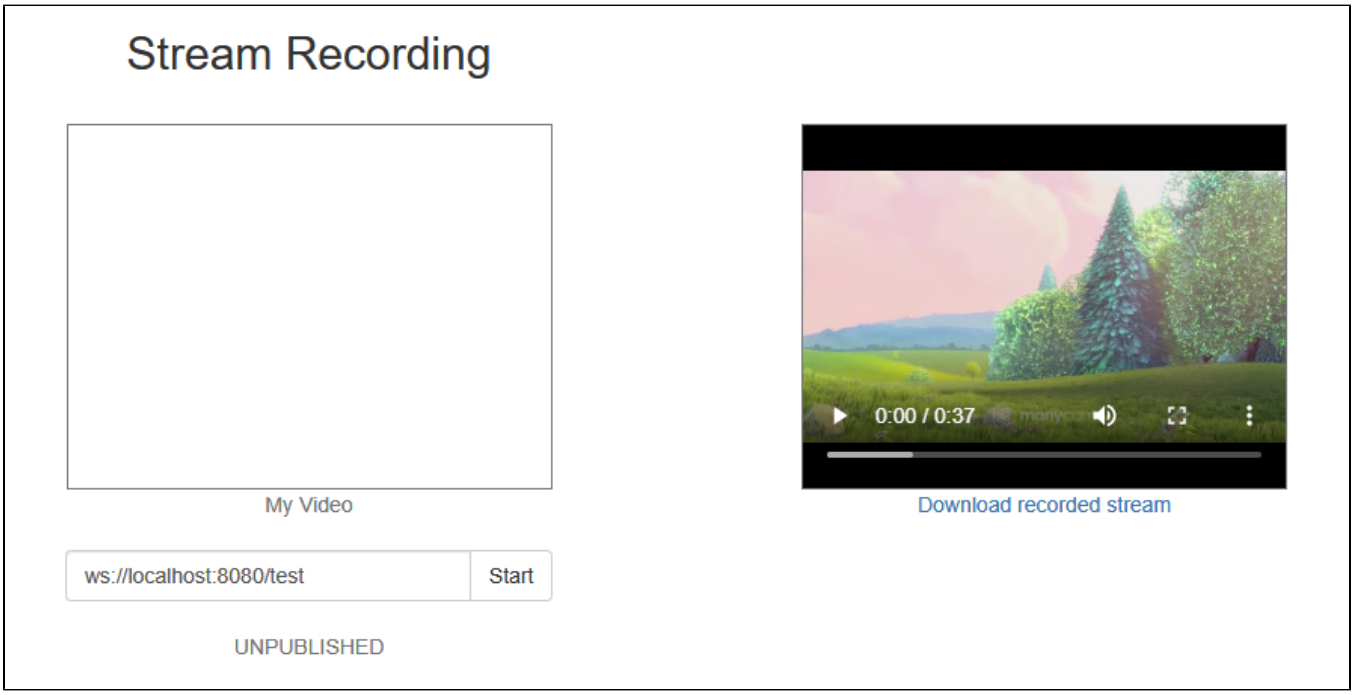
Stream Recording

Example of streamer with recording of published video stream

This streamer can be used to publish and record WebRTC streams on Web Call Server

On the screenshot below streaming from the client has been stopped.



While publishing, video from the camera is played in 'My Video'.
When streaming is stopped, recording of the video stream becomes available for downloading and playback in the player on the right side of the page.

Code of the example

The path to the source code of the example on WCS server is:

/usr/local/FlashphonerWebCallServer/client/examples/demo/streaming/stream_recording

recording.css - file with styles
recording.html - page of the streamer
recording.js - script providing functionality for the streamer

This example can be tested using the following address:

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

Here host is the address of the WCS server.

Analizing the code

To analyze the code, let's take the version of file recording.js with hash ecbadc3, which is available here and can be downloaded with corresponding build 2.0.212.

1. Initialization of the API

Flashphoner.init() code

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

2. Connection to server.

Flashphoner.createSession() code

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

3.Receiving the event confirming successful connection

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. Video streaming.

session.createStream(), publish() code
When stream is created, the following parameters are passed

- streamName - name of the stream
- localVideo - element, in which video from camera will be displayed
- true for parameter 'record' - to enable stream recording

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

5.Receiving the event confirming successful streaming

StreamStatusEvent PUBLISHING code

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UNPUBLISHED, function(stream) {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

6. Streaming stop

stream.stop() code

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

7.Receiving the event confirming successful streaming stop

StreamStatusEvent UNPUBLISHED code

On receiving the event, showDownloadLink() functionis called to form the record file download link. The record file name returned bystream.getRecordInfo() method passed to theshowDownloadLink() function.

```
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. Download link.

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();
    }
}
```

Stream recordings are saved to directory WCS_HOME/client/records.

When the link is formed:

- window.location.protocol is the protocol being used (HTTP or HTTPS)
- window.location.host is <address of the WCS server>:<port>
- name is name of the file with the recording