

В браузере с помощью Delight Player

- [Описание](#)
 - [Поддерживаемые платформы и браузеры](#)
 - [Технологии](#)
- [Использование WebRTC](#)
 - [Использование возможностей WebSDK](#)
 - [Тестирование](#)
 - [Пример кода страницы с плеером](#)
 - [Использование возможностей JavaScript и HTML5](#)
 - [Тестирование](#)
 - [Пример кода страницы с плеером](#)
- [Использование HLS](#)
 - [Тестирование](#)
 - [Пример кода страницы с плеером](#)
- [Известные проблемы](#)

Описание

Поток, опубликованный на WCS сервере, можно воспроизвести в браузерном VR-плеере, например, [Delight Player](#). Таким образом можно проигрывать поток в устройствах виртуальной и смешанной реальности, если на этом устройстве работает один из поддерживаемых браузеров. Отметим, что качество воспроизведения потока в устройстве VR будет тем выше, чем выше качество публикуемого потока.

Поддерживаемые платформы и браузеры

	Chrome	Firefox	Safari 11	Edge
Windows	+	+		+
Mac OS	+	+	+	
Android	+	+		
iOS	-	-	+	

Технологии

- WebRTC
- HLS

Использование WebRTC

Поток в Delight Player можно воспроизвести двумя способами:

1. С помощью WebSDK
2. С помощью только JavaScript и HTML5

Использование возможностей WebSDK

Чтобы воспроизвести поток по WebRTC в Delight Player или любом другом стороннем JavaScript плеере, видеоэлемент страницы, в котором будет воспроизводиться поток, передается параметром `remoteVideo` в функцию `WebSDKsession.createStream()`

`session.createStream()` код

```
session.createStream({
  name: document.getElementById('playStream').value,
  display: display,
  remoteVideo: video
})
...
```

Тестирование

1. Для теста возьмем:

- WCS сервер
- веб-приложение [Media Devices](#) для публикации потока в высоком разрешении
- VR-плеер [Delight](#) для воспроизведения потока

2. Установим разрешение публикуемого потока 1920x1080

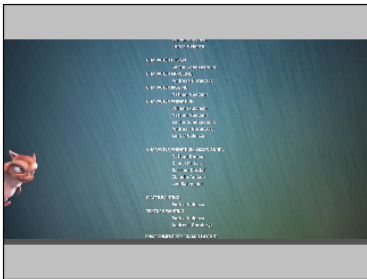
Screen share	<input type="checkbox"/> off	
Size	<input type="text" value="1920"/>	<input type="text" value="1080"/>
FPS	<input type="text" value="30"/>	
Bitrate	min <input type="text" value="0"/>	max <input type="text" value="0"/>

3. Укажем в поле WCS имя потока `wss://test1.flashphoner.com:8443/test` и нажмем Start для публикации

Media Devices

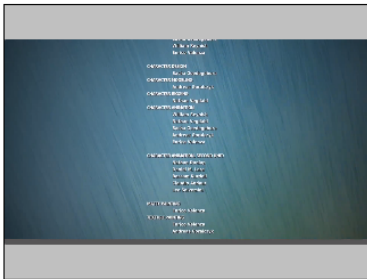
Video stats
Bytes sent: 8460431
Packets sent: 7719
Frames encoded: 1274
Audio stats
Bytes sent: 200204
Packets sent: 2237

Local



1920x1080

Preview



1920x1080

Video stats
Bytes received: 8422719
Packets received: 6870
Frames decoded: 1247
Audio stats
Bytes received: 197275
Packets received: 2202

WCS

PUBLISHING

4. Воспроизводим поток в VR-плеере



Угол зрения в браузере на ПК можно менять мышью, на iOS или специализированном VR-устройстве угол зрения меняется в зависимости от положения в пространстве.

Пример кода страницы с плеером

1. Объявление видеоэлемента для воспроизведения потока, поля ввода имени потока и кнопок запуска и остановки воспроизведения

```
<div style="width: 50%;" id="display">
  <dl8-live-video id="remoteVideo" format="STEREO_TERPON">
    <source>
  </dl8-live-video>
</div>
  <input class="form-control" type="text" id="playStream" placeholder="Stream Name">
  <button id="playBtn" type="button" class="btn btn-default" disabled>Play</button>
  <button id="stopBtn" type="button" class="btn btn-default" disabled>Stop</button>
```

2. Обработка события готовности плеера к воспроизведению

```
document.addEventListener('x-dl8-evt-ready', function () {
  dl8video = document.getElementById('remoteVideo');
  $('#playBtn').prop('disabled', false).click(function() {
    playStream();
  });
});
```

3. Установка соединения с сервером и создание потока

```

        var video = dl8video.contentElement;
        Flashphoner.createSession({urlServer: url}).on(SESSION_STATUS.ESTABLISHED, function
(session) {
            var session = Flashphoner.getSessions()[0];
            session.createStream({
                name: document.getElementById('playStream').value,
                display: display,
                remoteVideo: video
            }).on(STREAM_STATUS.PLAYING, function (stream) {
                ...
            }).play();
        })
    }
}

```

4. Запуск воспроизведения в VR-плеере и обработка нажатия кнопки остановки воспроизведения

```

        ...
    }).on(STREAM_STATUS.PLAYING, function (stream) {
        dl8video.start();
        $('#stopBtn').prop('disabled', false).click(function() {
            $('#playBtn').prop('disabled', false);
            $('#stopBtn').prop('disabled', true);
            stream.stop();
            dl8video.exit();
        });
    }).play();
}
}

```

Полный код примера страницы с VR-плеером

Code

```
<!DOCTYPE html>
<html>
  <head>
    <title>WebRTC Delight</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <script type="text/javascript" src="../../../../flashphoner.js"></script>
    <script type="text/javascript" src="../../dependencies/jquery/jquery-1.12.0.js"></script>
    <script type="text/javascript" src="../../dependencies/js/utls.js"></script>
    <script src="dl8-66b250447635476dl123a44a391c80b09887e831e.js" async></script>
    <meta name="dl8-custom-format" content='{ "name": "STEREO_TERPON", "base": "STEREO_MESH", "params": { "uri":
"03198702.json" } }'>
  </head>
  <body>
    <div style="width: 50%;" id="display">
      <dl8-live-video id="remoteVideo" format="STEREO_TERPON">
        <source>
      </dl8-live-video>
    </div>
    <input class="form-control" type="text" id="playStream" placeholder="Stream Name">
    <button id="playBtn" type="button" class="btn btn-default disabled>Play</button>
    <button id="stopBtn" type="button" class="btn btn-default disabled>Stop</button>
    <script>
      Flashphoner.init({flashMediaProviderSwfLocation: '../../../../media-provider.swf'});
      var SESSION_STATUS = Flashphoner.constants.SESSION_STATUS;
      var STREAM_STATUS = Flashphoner.constants.STREAM_STATUS;
      var STREAM_STATUS_INFO = Flashphoner.constants.STREAM_STATUS_INFO;
      var playBtn = document.getElementById('playBtn');
      var display = document.getElementById('display');
      var dl8video = null;
      var url = setURL();
      document.addEventListener('x-dl8-evt-ready', function () {
        dl8video = document.getElementById('remoteVideo');
        $('#playBtn').prop('disabled', false).click(function() {
          playStream();
        });
      });
      function playStream() {
        $('#playBtn').prop('disabled', true);
        $('#stopBtn').prop('disabled', false);
        var video = dl8video.contentElement;
        Flashphoner.createSession({urlServer: url}).on(SESSION_STATUS.ESTABLISHED, function
(session) {
          var session = Flashphoner.getSessions()[0];
          session.createStream({
            name: document.getElementById('playStream').value,
            display: display,
            remoteVideo: video
          }).on(STREAM_STATUS.PLAYING, function (stream) {
            dl8video.start();
            $('#stopBtn').prop('disabled', false).click(function() {
              $('#playBtn').prop('disabled', false);
              $('#stopBtn').prop('disabled', true);
              stream.stop();
              dl8video.exit();
            });
          }).play();
        })
      }
    </script>
  </body>
</html>
```

Чтобы воспроизвести поток по WebRTC в Delight Player или любом другом стороннем JavaScript плеере, на странице создается псевдоэлемент для вывода потока

```
var mockRemoteDisplay = $('<div></div>');
var mockRemoteVideo = $('<video></video>', {id: 'mock-REMOTE_CACHED_VIDEO'});
mockRemoteDisplay.append(mockRemoteVideo);
```

Псевдоэлемент `mockRemoteDisplay` передается параметром `display` в функцию `WebSDKsession.createStream()`, а элемент `mockRemoteVideo` передается как источник потока в VR-плеер

`session.createStream()` код

```
session.createStream({
  name: $('#streamName').val(),
  display: mockRemoteDisplay.get(0)
}).on(STREAM_STATUS.PLAYING, function (stream) {
  var srcObject = mockRemoteVideo.get(0).srcObject;
  video.srcObject = srcObject;
  dl8video.start();
  ...
}).play();
```

Тестирование

1. Для теста возьмем:

- WCS сервер
- веб-приложение [Media Devices](#) для публикации потока в высоком разрешении
- VR-плеер [Delight](#) для воспроизведения потока

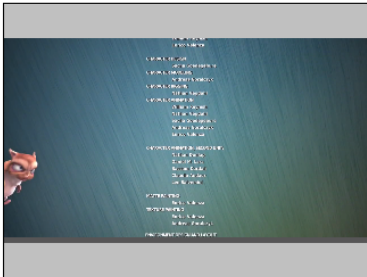
2. Установим разрешение публикуемого потока 1920x1080

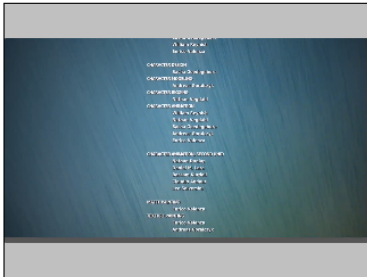
Screen share	<input type="checkbox"/> off	
Size	<input type="text" value="1920"/>	<input type="text" value="1080"/>
FPS	<input type="text" value="30"/>	
Bitrate	min <input type="text" value="0"/>	max <input type="text" value="0"/>

3. Укажем в поле WCS имя потока `wss://test1.flashphoner.com:8443/test1` нажмем Start для публикации

Media Devices

Video stats
 Bytes sent: 8460431
 Packets sent: 7719
 Frames encoded: 1274
Audio stats
 Bytes sent: 200204
 Packets sent: 2237

Local

 1920x1080

Preview

 1920x1080

Video stats
 Bytes received: 8422719
 Packets received: 6870
 Frames decoded: 1247
Audio stats
 Bytes received: 197275
 Packets received: 2202

WCS

wss://test1.flashphoner.com:8443/
 PUBLISHING

4. Воспроизводим поток в VR-плеере



Пример кода страницы с плеером

1. Объявление видеоэлемента для воспроизведения потока, поля ввода имени потока и кнопок запуска и остановки воспроизведения

```
<div style="width: 50%;">
  <dl8-live-video id="remoteVideo" format="STEREO_TERPON" muted="true">
    <source>
  </dl8-live-video>
</div>
<input class="form-control" type="text" id="streamName" placeholder="Stream Name">
<button id="playBtn" type="button" class="btn btn-default" disabled>Play</button>
<button id="stopBtn" type="button" class="btn btn-default" disabled>Stop</button>
```

2. Обработка события готовности плеера к воспроизведению

```
document.addEventListener('x-dl8-evt-ready', function () {
    dl8video = $('#remoteVideo').get(0);
    $('#playBtn').prop('disabled', false).click(function() {
        publishStream();
    });
});
```

3. Создание псевдоэлементов для воспроизведения потока

```
var mockRemoteDisplay = $('<div></div>');
var mockRemoteVideo = $('<video></video>', {id: 'mock-REMOTE_CACHED_VIDEO'});
mockRemoteDisplay.append(mockRemoteVideo);
```

4. Установка соединения с сервером и создание потока

```
(session) {
    var video = dl8video.contentElement;
    Flashphoner.createSession({urlServer: url}).on(SESSION_STATUS.ESTABLISHED, function
    var session = Flashphoner.getSessions()[0];
    session.createStream({
        name: $('#streamName').val(),
        display: mockRemoteDisplay.get(0)
    }).on(STREAM_STATUS.PLAYING, function (stream) {
        ...
    }).play();
})
```

5. Запуск воспроизведения в VR-плеере и обработка нажатия кнопки остановки воспроизведения

```
...
}).on(STREAM_STATUS.PLAYING, function (stream) {
    var srcObject = mockRemoteVideo.get(0).srcObject;
    video.srcObject = srcObject;
    dl8video.start();
    mockRemoteVideo.get(0).pause();
    mockRemoteVideo.get(0).srcObject = null;
    $('#stopBtn').prop('disabled', false).click(function() {
        stream.stop();
        $('#playBtn').prop('disabled', false);
        $('#stopBtn').prop('disabled', true);
        dl8video.exit();
    });
}).play();
```

Полный код примера страницы с VR-плеером

Code


```

<!DOCTYPE html>
<html>
  <head>
    <title>WebRTC Delight</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <script type="text/javascript" src="../../../../flashphoner.js"></script>
    <script type="text/javascript" src="../../dependencies/jquery/jquery-1.12.0.js"></script>
    <script type="text/javascript" src="../../dependencies/js/Utils.js"></script>
    <script src="dl8-66b250447635476d123a44a391c80b09887e831e.js" async></script>
    <meta name="dl8-custom-format" content='{ "name": "STEREO_TERPON", "base": "STEREO_MESH", "params": { "uri":
"03198702.json"} }'>
  </head>
  <body>
    <div style="width: 50%;">
      <dl8-live-video id="remoteVideo" format="STEREO_TERPON" muted="true">
        <source>
      </dl8-live-video>
    </div>
    <input class="form-control" type="text" id="streamName" placeholder="Stream Name">
    <button id="playBtn" type="button" class="btn btn-default" disabled>Play</button>
    <button id="stopBtn" type="button" class="btn btn-default" disabled>Stop</button>
    <script>
      Flashphoner.init({flashMediaProviderSwfLocation: '../../media-provider.swf'});
      var SESSION_STATUS = Flashphoner.constants.SESSION_STATUS;
      var STREAM_STATUS = Flashphoner.constants.STREAM_STATUS;
      var STREAM_STATUS_INFO = Flashphoner.constants.STREAM_STATUS_INFO;
      var playBtn = $('#playBtn').get(0);
      var dl8video = null;
      var url = setURL();
      document.addEventListener('x-dl8-evt-ready', function () {
        dl8video = $('#remoteVideo').get(0);
        $('#playBtn').prop('disabled', false).click(function() {
          publishStream();
        });
      });

      var mockRemoteDisplay = $('#div></div>');
      var mockRemoteVideo = $('#<video></video>', {id: 'mock-REMOTE_CACHED_VIDEO'});
      mockRemoteDisplay.append(mockRemoteVideo);
      function publishStream() {
        $('#playBtn').prop('disabled', true);
        $('#stopBtn').prop('disabled', false);
        var video = dl8video.contentElement;
        Flashphoner.createSession({urlServer: url}).on(SESSION_STATUS.ESTABLISHED, function
(session) {

          var session = Flashphoner.getSessions()[0];
          session.createStream({
            name: $('#streamName').val(),
            display: mockRemoteDisplay.get(0)
          }).on(STREAM_STATUS.PLAYING, function (stream) {
            var srcObject = mockRemoteVideo.get(0).srcObject;
            video.srcObject = srcObject;
            dl8video.start();
            mockRemoteVideo.get(0).pause();
            mockRemoteVideo.get(0).srcObject = null;
            $('#stopBtn').prop('disabled', false).click(function() {
              stream.stop();
              $('#playBtn').prop('disabled', false);
              $('#stopBtn').prop('disabled', true);
              dl8video.exit();
            });
          });
        }).play();
      })
    </script>
  </body>
</html>

```

Использование HLS

В тех случаях, когда воспроизведением потока в Delight Player по WebRTC возникают проблемы, можно проиграть поток по HLS

Тестирование

1. Для теста возьмем:

- WCS сервер
- веб-приложение [Media Devices](#) для публикации потока в высоком разрешении
- VR-плеер [Delight](#) для воспроизведения потока

2. Установим разрешение публикуемого потока 1920x1080

Screen share

off

Size

1920

1080

FPS

30

Bitrate

min

0

max

0

3. Укажем в поле WCS имя потока `wss://test1.flashphoner.com:8443/test1` нажмем Start для публикации

Media Devices

Video stats

Bytes sent: 8460431

Packets sent: 7719

Frames encoded: 1274

Audio stats

Bytes sent: 200204

Packets sent: 2237

Local

1920x1080

WCS

wss://test1.flashphoner.com:8443/test1

PUBLISHING

Preview

1920x1080

Video stats

Bytes received: 8422719

Packets received: 6870

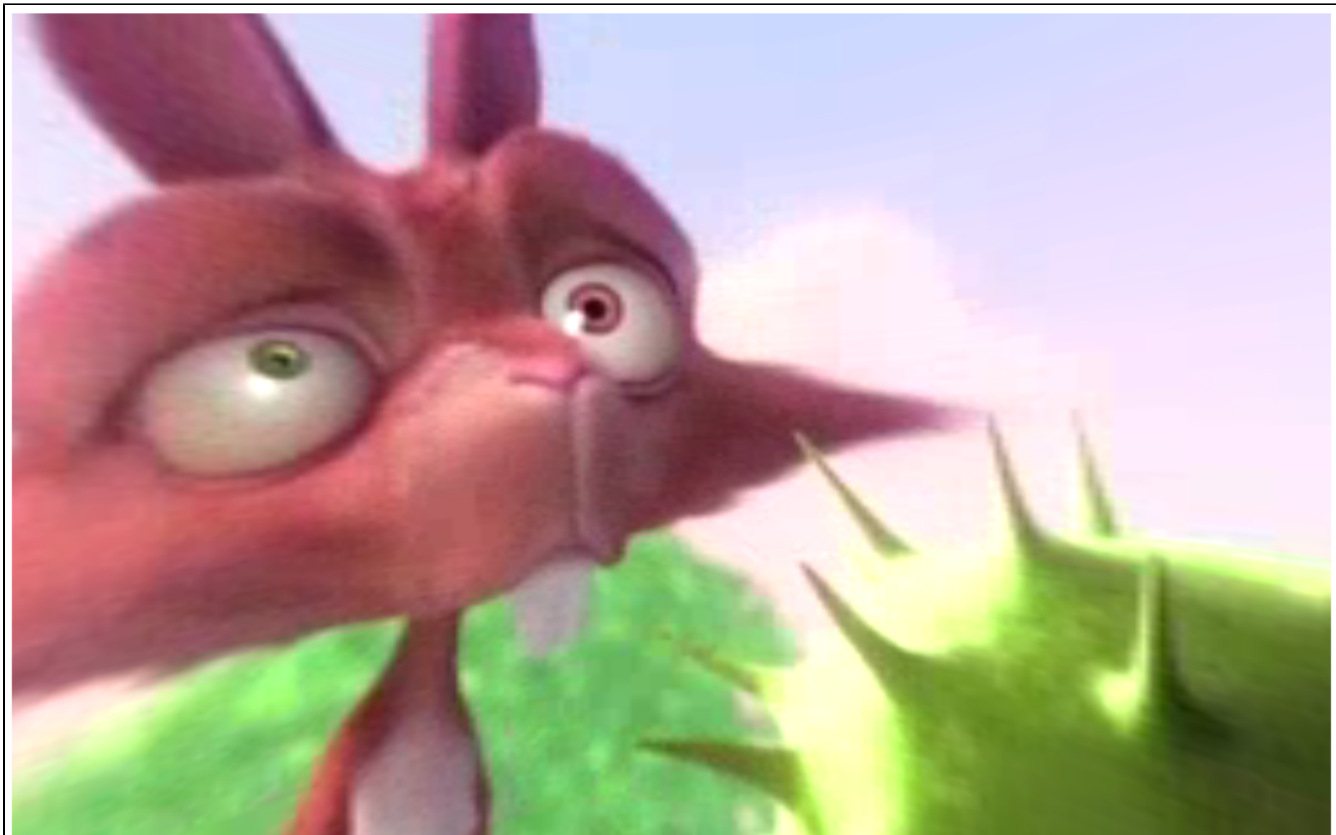
Frames decoded: 1247

Audio stats

Bytes received: 197275

Packets received: 2202

4. Воспроизводим поток в VR-плеере



test

Play

Stop

Пример кода страницы с плеером

1. Объявление видеоэлемента для воспроизведения потока, поля ввода имени потока и кнопок запуска и остановки воспроизведения

```
<div style="width: 50%;" id="display">
  <dl8-live-video id="remoteVideo" format="MONO_360">
    <source type="application/x-mpegurl" id="hlsSource" />
  </dl8-live-video>
</div>
<input class="form-control" type="text" id="playStream" placeholder="Stream Name">
<button id="playBtn" type="button" class="btn btn-default" disabled>Play</button>
<button id="stopBtn" type="button" class="btn btn-default" disabled>Stop</button>
```

2. Обработка события готовности плеера к воспроизведению

```
document.addEventListener('x-dl8-evt-ready', function () {
  dl8video = document.getElementById('remoteVideo');
  $('#playBtn').prop('disabled', false).click(playStream);
});
```

3. Получение URL сервера для воспроизведения HLS

```
var hlsUrl = getHLSUrl();
```

4. Запуск воспроизведения в VR-плеере и обработка нажатия кнопки остановки воспроизведения

```

...
var video = dl8video.contentElement;
var streamName = document.getElementById('playStream').value;
$('#hlsSource').attr("src",hlsUrl + "/" + streamName + "/" + streamName + ".m3u8");
dl8video.start();
$('#stopBtn').prop('disabled', false).click(function() {
    $('#playBtn').prop('disabled', false);
    $('#stopBtn').prop('disabled', true);
    dl8video.exit();
});

```

Полный код примера страницы с VR-плеером

Code

```

<!DOCTYPE html>
<html>
  <head>
    <title>WebRTC Delight</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <script type="text/javascript" src="../../../../flashphoner.js"></script>
    <script type="text/javascript" src="../../dependencies/jquery/jquery-1.12.0.js"></script>
    <script type="text/javascript" src="../../dependencies/js/utils.js"></script>
    <script src="dl8-66b250447635476d123a44a391c80b09887e831e.js" async></script>
  </head>
  <body>
    <div style="width: 50%;" id="display">
      <dl8-live-video id="remoteVideo" format="MONO_360">
        <source type="application/x-mpegurl" id="hlsSource"/>
      </dl8-live-video>
    </div>
    <input class="form-control" type="text" id="playStream" placeholder="Stream Name">
    <button id="playBtn" type="button" class="btn btn-default disabled>Play</button>
    <button id="stopBtn" type="button" class="btn btn-default disabled>Stop</button>

    <script>
      var playBtn = document.getElementById('playBtn');
      var display = document.getElementById('display');
      var dl8video = null;
      var hlsUrl = getHLSUrl();
      document.addEventListener('x-dl8-evt-ready', function () {
        dl8video = document.getElementById('remoteVideo');
        $('#playBtn').prop('disabled', false).click(playStream);
      });

      function playStream() {
        $('#playBtn').prop('disabled', true);
        $('#stopBtn').prop('disabled', false);
        var video = dl8video.contentElement;
        var streamName = document.getElementById('playStream').value;
        $('#hlsSource').attr("src",hlsUrl + "/" + streamName + "/" + streamName + ".m3u8");
        dl8video.start();
        $('#stopBtn').prop('disabled', false).click(function() {
          $('#playBtn').prop('disabled', false);
          $('#stopBtn').prop('disabled', true);
          dl8video.exit();
        });
      }
    </script>
  </body>
</html>

```

Известные проблемы

1. Поток не воспроизводится в Delight Player по WebRTCили воспроизводится с фризами.

Симптомы: поток в Delight Player не воспроизводится вообще (MS Edge) либо воспроизводится с постоянными фризами (iOS Safari)

Решение: использовать HLS для воспроизведения потока

2. VR-отображение не работает при воспроизведении потока в Delight Player по HLS в MS Edge на Windows 10 Mobile.

Симптомы: поток в Delight Player воспроизводится по HLS, но картинка плоская

Решение: использовать устройство на актуальной операционной системе с поддержкой большого количества браузеров.

3. Поток не воспроизводится в браузере Safari по HLS.

Симптомы: поток в Delight Player не воспроизводится, индикатор загрузки отображает 99%, затем черный экран, либо отображается ошибка CORS

Решение: использовать [nginx в качестве обратного прокси](#) для воспроизведения потока по HLS в Safari