

Player


Пример плеера

Данный плеер может использоваться для воспроизведения любого типа потока с Web Call Server:

- RTSP
- WebRTC
- RTMP
- RTMFP
- SIP

Ниже представлен пример воспроизведения RTSP-потока в web-плеере.

Player



WCS URL

Stream

PLAYING

Код примера

Код данного примера находится на сервере по следующему пути:

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

player.css - файл стилей

player.html - страница с плеером

player.js - скрипт, обеспечивающий работу плеера

Тестировать данный пример можно по следующему адресу:

<https://host:8888/client2/examples/demo/streaming/player/player.html>

Здесь host - адрес вашего WCS-сервера.

Работа с кодом примера

Для разбора кода возьмем версию файла player.js с хешем 66сс393, которая находится [здесь](#) и доступна для скачивания в соответствующей сборке [0.5.28.2753.133](#).

1. Инициализация API.

Flashphoner.init() [code](#)

```
Flashphoner.init({
  flashMediaProviderSwfLocation: '../../../media-provider.swf',
  receiverLocation: '../../../dependencies/websocket-player/WSReceiver2.js',
  decoderLocation: '../../../dependencies/websocket-player/video-worker2.js',
  preferredMediaProvider: mediaProvider
});
```

2. Подключение к серверу.

Flashphoner.createSession() [code](#)

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

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
  playStream(session);
}).on(SESSION_STATUS.DISCONNECTED, function(){
  ...
}).on(SESSION_STATUS.FAILED, function(){
  ...
});
```

4. Воспроизведение видеопотока.

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

```
stream = session.createStream(options).on(STREAM_STATUS.PENDING, function (stream) {
  ...
});
stream.play();
```

При создании передаются:

- streamName - имя видеопотока
- remoteVideo - div-элемент, в котором будет отображаться видео
- разрешение, к которому должна быть транскодирована картинка

```

var options = {
  name: streamName,
  display: remoteVideo,
  flashShowFullScreenButton: true
};
if (Flashphoner.getMediaProviders()[0] === "MSE" && mseCutByIFrameOnly) {
  options.mediaConnectionConstraints = {
    cutByIFrameOnly: mseCutByIFrameOnly
  }
}
if (resolution_for_wsplayer) {
  options.playWidth = resolution_for_wsplayer.playWidth;
  options.playHeight = resolution_for_wsplayer.playHeight;
} else if (resolution) {
  options.playWidth = resolution.split("x")[0];
  options.playHeight = resolution.split("x")[1];
}

```

5. Получение от сервера события, подтверждающего успешное воспроизведение потока

StreamStatusEvent PLAYING [code](#)

```

stream = session.createStream(options).on(STREAM_STATUS.PENDING, function (stream) {
  ...
}).on(STREAM_STATUS.PLAYING, function (stream) {
  $("#preloader").hide();
  setStatus(stream.status());
  onStart(stream);
}).on(STREAM_STATUS.STOPPED, function() {
  ...
}).on(STREAM_STATUS.FAILED, function() {
  ...
});
stream.play();

```

6. Остановка воспроизведения видеопотока.

stream.stop() [code](#)

```

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

```

7. Получение от сервера события, подтверждающего успешную остановку воспроизведения потока

StreamStatusEvent STOPPED [code](#)

```

stream = session.createStream(options).on(STREAM_STATUS.PENDING, function (stream) {
  ...
}).on(STREAM_STATUS.STOPPED, function () {
  $("#preloader").hide();
  setStatus(STREAM_STATUS.STOPPED);
  onStop();
}).on(STREAM_STATUS.FAILED, function() {
  ...
}).play();

```

8. Регулировка громкости воспроизведения

stream.unmuteRemoteAudio(),stream.setVolume(currentVolumeValue) [code](#)

```
$("#volumeControl").slider({
  range: "min",
  min: 0,
  max: 100,
  value: currentVolumeValue,
  step: 10,
  animate: true,
  slide: function(event, ui) {
    //WCS-2375. fixed autoplay in ios safari
    stream.unmuteRemoteAudio();
    currentVolumeValue = ui.value;
    stream.setVolume(currentVolumeValue);
  }
}).slider("disable");
```