

HLS Native Player

Пример конвертации видеопотока в HLS и отображения его в браузере, поддерживающем HLS

Данный плеер демонстрирует возможности WCS по преобразованию опубликованного на сервере потока в HLS и воспроизведению его в браузере. Нарезка потока в HLS запускается автоматически, при обращении к потоку, опубликованному на сервере, по HLS URL, например, для потока на рисунке ниже <https://demo.flashphoner.com:8445/test/test.m3u8>

HLS Native Player Minimal


WCS

Stream

Auth

Key	Value
-----	-------

Stop



Код примера

Код данного примера находится на сервере по следующему пути:

```
/usr/local/FlashphonerWebCallServer/client2/examples/demo/streaming/hls-js-player
```

hls-native.css - файл стилей страницы с плеером

hls-native.html - страница с плеером

hls-native.js - скрипт, обеспечивающий запуск плеера

Тестировать данный пример можно по следующему адресу:

```
https://host:8888/client2/examples/demo/streaming/hls-js-player/hls-js-player.html
```

Здесь host - адрес вашего WCS-сервера.

Работа с кодом примера

Для разбора кода возьмем версию файла hls-native.js с хешем 51703a2, которая находится [здесь](#) и доступна для скачивания в соответствующей сборке [0.5.28.2753.141](#).

1. Определение HLS URL сервера

getHLSUrl() [code](#)

```
function initPage() {  
    $("#header").text("HLS Native Player Minimal");  
    $("#urlServer").val(getHLSUrl());  
    ...  
}
```

2. Настройка div элемента для передачи плееру

[code](#)

Плееру передается имя div-элемента, в котором должен быть проигран поток.

```
function initPage() {
    ...
    remoteVideo = document.getElementById('remoteVideo');
    remoteVideo.style = "background-color: lightgrey;";
}
```

3. Определение имени потока (должен быть опубликован на сервере)

`encodeURIComponent()` [code](#)

```
function playBtnClick() {
    if (validateForm()) {
        var streamName = $('#playStream').val();
        streamName = encodeURIComponent(streamName);
        ...
    }
}
```

4. Формирование URL HLS-потока

[code](#)

Если указаны ключ и токен авторизации, они будут включены в URL потока

```
function playBtnClick() {
    if (validateForm()) {
        ...
        var videoSrc = $("#urlServer").val() + '/' + streamName + '/' + streamName + '.m3u8';
        var key = $('#key').val();
        var token = $("#token").val();
        if (key.length > 0 && token.length > 0) {
            videoSrc += "?" + key + "=" + token;
        }
        ...
    }
}
```

5. Запуск плеера

[code](#)

Если браузер не поддерживает воспроизведение HLS, плеер не запустится, будет выведено предупреждение

```
function playBtnClick() {
    if (validateForm()) {
        ...
        if (remoteVideo.canPlayType('application/vnd.apple.mpegurl')) {
            remoteVideo.src = videoSrc;
            remoteVideo.addEventListener('loadedmetadata', function() {
                console.log("Play native HLS");
                remoteVideo.play();
                onStarted();
            });
        }
        else {
            $("#notifyFlash").text("Your browser doesn't support native HLS playback");
        }
    }
}
```

6. Остановка воспроизведения

code

```
function stopBtnClick() {
  if (remoteVideo != null) {
    console.log("Stop HTML5 player");
    remoteVideo.pause();
    remoteVideo.currentTime = 0;
    remoteVideo.removeAttribute('src');
    remoteVideo.load();
  }
  onStopped();
}
```