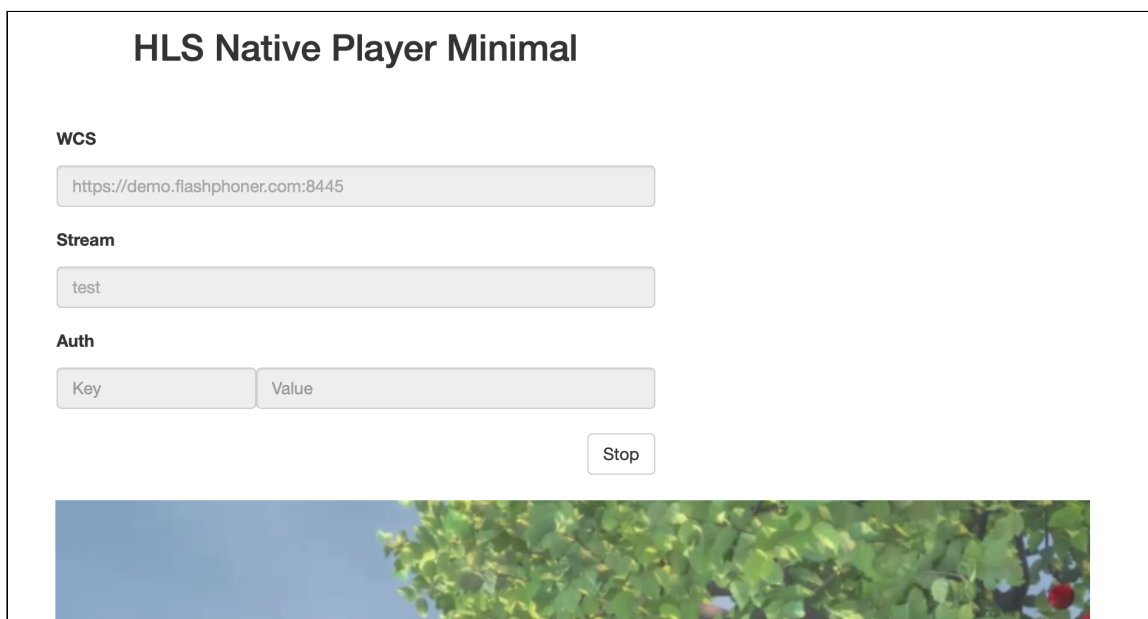


HLS Native Player

Example of stream conversion to HLS and playing it in browser supporting HLS natively

The player shows how to convert stream published on WCS server to HLS and play it in browser. HLS stream cut starts automatically when stream is requested by HLS URL, for example `https://demo.flashphoner.com:8445/test/test.m3u8` on the screenshot below



The code of the example

The source code can be accessed on server by the following path:

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

- `hls-native.css` - player page styles file
- `hls-native.html` - player page
- `hls-native.js` - player launch script

The example can be tested using the following URL:

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

Where host is WCS server address

Analyzing the code

To analyze the code get `hls-native.js` file version with hash `ecbadc3` which is available [here](#) and can be downloaded in build [2.0.212](#).

1. A server HLS URL detection

`getHLSUrl()` [code](#)

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

2. `div` element setup to pass to the player

[code](#)

A div element for stream playback is passed to player

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

3. Stream name detection (the stream should be published to server)

`encodeURIComponent()` [code](#)

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

4. HLS stream URL forming

[code](#)

If authentication key and token are set, they will be included to stream URL

```
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. Player starting

code

If browser does not support HLS playback natively, player will not be launched and a warning will be displayed

```

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();
                onStart();
            });
        }
        else {
            $("#notifyFlash").text("Your browser doesn't support native HLS
playback");
        }
    }
}
}

```

6. Playback stopping

code

```

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

```