

# Автоматическое восстановление воспроизведения потока

## Пошаговые примеры кода

Если воспроизведение WebRTC потока по какой-либо причине остановилось, можно настроить автоматическое переподключение к этому потоку, модифицировав скрипт примера Player:

1. Добавляем в начало скрипта переменные для настройки автоматического переподключения

```
var restart = eval(getUrlParam("restart")) || false;
// Settings to restart playback if failed
var restartTimeout = 3000; //ms
var restartMaxTimes = 100; //will try to restart playback for 5 minutes
var restartCount = 0;
var restartTimerId;
```

В данном примере, попытка повторного подключения к потоку будет производиться каждые 3 секунды, всего будет сделано 100 попыток. Таким образом, максимально возможное время ожидания возобновления потока составляет 5 минут, после чего попытки восстановления прекращаются

2. В обработчик события `STREAM_STATUS.PLAYING` добавляем вызов функции `clearRestart()` для сброса очередного таймера переподключения, если он активен

```
function onStarted(stream) {
    ...
    clearRestart();
}
```

3. В обработчик события `SESSION_STATUS.FAILED` добавляем вызов функции `tryToRestart()` запуск таймера переподключения

```
function start() {
    var url = $('#url').val();
    ...
    Flashphoner.createSession({urlServer:
url}).on(SESSION_STATUS.ESTABLISHED, function(session){
        ...
    }).on(SESSION_STATUS.DISCONNECTED, function(){
        ...
    }).on(SESSION_STATUS.FAILED, function(){
        setStatus(SESSION_STATUS.FAILED);
    });
}
```

```

        onStoped();
        tryToRestart();
    });
}

```

4. В обработчик события `STREAM_STATUS.STOPPED` добавляем вызов функции `clearRestart()` для сброса очередного таймера переподключения, если он активен

```

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

```

5. В обработчик события `STREAM_STATUS.FAILED` добавляем вызов функции `tryToRestart()` для запуска таймера переподключения

```

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

```

6. Добавляем в скрипт функцию запуска таймера переподключения к потоку

```
function tryToRestart() {
    if (restart) {
        restartTimerId = setTimeout(function(){
            if (stream && (stream.status() != STREAM_STATUS.PLAYING) &&
restartCount < restartMaxTimes){
                $("#playBtn").click();
                restartCount++;
            }
            if (restartCount >= restartMaxTimes) {
                console.log("Tried to restart playback for
"+restartMaxTimes+" times with "+restartTimeout+" ms interval,
cancelled");
            }
        },restartTimeout);
    }
}
```

7. Добавляем в скрипт функцию сброса таймера переподключения

```
function clearRestart() {
    if (restart) {
        clearTimeout(restartTimerId);
        restartCount = 0;
    }
}
```

## Готовые примеры

Готовый пример Player на базе WebSDK 2.0.228 можно скачать здесь

[player\\_restart.tar.gz](#)

Готовый пример Embed Player на базе WebSDK 2.0.228 можно скачать здесь

[embed\\_player\\_restart.tar.gz](#)