

How to reconnect to a stream automatically

- [Ready to use samples](#)

If WebRTC stream playback is stopped by some reason, an automatic reconnection can be implemented by tweaking Player example code as follows:

1. Add the variables to configure automatic reconnection to the script beginning

```
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;
```

In this example, player will try to reconnect to the stream every 3 seconds until 100 tries are expired. Therefore, maximum time to wait for the stream to be restored is 5 minutes, then automatic reconnection is cancelled

2. Add `clearRestart()` function call to clear reconnection timer if enabled to `STREAM_STATUS.PLAYING` handler

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

3. Add `tryToRestart()` function call to enable reconnection timer to `SESSION_STATUS.FAILED` handler

```
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);
        onStopped();
        tryToRestart();
    });
}
```

4. Add `clearRestart()` function call to clear reconnection timer if enabled to `STREAM_STATUS.STOPPED` handler

```
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();
        onStopped();
    }).on(STREAM_STATUS.FAILED, function(stream) {
        ...
    }).on(STREAM_STATUS.NOT_ENOUGH_BANDWIDTH, function(stream){
        ...
    });
    stream.play();
}
```

5. Add `tryToRestart()` function call to enable reconnection timer to `STREAM_STATUS.FAILED` handler

```
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);
        onStopped();
        tryToRestart();
    }).on(STREAM_STATUS.NOT_ENOUGH_BANDWIDTH, function(stream){
        ...
    });
    stream.play();
}
```

6. Add the function to enable reconnection timer

```
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. Add the function to clear reconnection timer

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

Ready to use samples

Ready to use Player example based on WebSDK [2.0.228](#) can be downloaded [here](#)



player_restart.tar.gz

Ready to use Embed Player example based on WebSDK [2.0.228](#) can be downloaded here



embed_player_restart.tar.gz