



Attention! It is strongly recommended to read next document: [JavaScript API for controlling IPTV devices](#)

Minimal HTML page. Start playing the stream.

Example 1.

Example of minimal HTML page which load and perform the stream's playing (RTP in this case).

[Code](#)

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
<title></title>
<script>
var stb = gSTB;
function init() {
    stb.InitPlayer()
    stb.Play("udp://239.1.1.1:1234")
}
function getkeydown(e) {
    ec = e.keyCode
    ew = e.which
    es = e.shiftKey
    pat = /^(\S+)_ (\S+)/
    switch (ew) {
    case 114: // Play
        stb.Play("udp://239.1.1.1:1234")
        break;
    case 115: // Stop
        stb.Stop()
        break;
    }
}
</script>
<body onload="init()" onKeyDown="getkeydown(event)">
</body>
```

</html>

Example 2.

Minimal HTML page. Starting playback stream **RTSP**, with preset parameters RTSP-server.

RTSP-server:

- Server type: 4 - Live555;
- Control flags: 32 (0x20) - Use UDP transport to send video:
- 1 - switch on the keep-alive mode.

[More details about setting parameters RTSP-server](#) (Section stb.SetupRTSP page 35)

[Code](#)

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
    <title></title>
    <script>
      var stb = gSTB;
      function init() {
        stb.InitPlayer();
        stb.SetupRTSP(4, 33); //Set parameters of RTSP-server
        stb.Play("rtsp://example.com/example.stream");
      }
      function getkeydown(e) {
        ec = e.keyCode
        ew = e.which
        es = e.shiftKey
        pat = /^(\\S+)_ (\\S+)/
        switch (ew) {
          case 114: // Play
            stb.Play("rtsp://example.com/example.stream")
            break;
          case 115: // Stop
            stb.Stop()
            break;
        }
      }
    </script>
    <body onload="init()" onKeyDown="getkeydown(event)">
  </body>
</html>
```

Event Model

Read more: [JS API MAG200 \(Rev 1.20\) - Event Model in JavaScript - page 83](#)

Example 3.

The stream will be requests again if there is “event 1”.

[Code](#)

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
    <title></title>
    <script>
      var stbEvent = {
        onEvent : function (event) {
          stb.Debug(event);
          if (event == 1) {
            stb.Play("udp://239.1.1.1:1234")
          }
        },
        event : 0
      }
      var stb = gSTB;
      function init() {
        stb.InitPlayer()
        stb.Play("udp://239.1.1.1:1234")
      }
    </script>
    <body onload="init()" onKeyDown="getkeydown(event)">
  </body>
</html>
```

Example 4.

The stream will be requests again if there are “**event 1**” and “**event 5**”. In current time <var check_time> the file <var fileName> begins to play from the root directory of USB-storage which connected to STB. When the file stops to play (or there is no file) then the stream <var streamURL> requests to play.

- **var check_time** - Time in 24h format
- **var streamURL** - Stream address

- **var fileName** - File name on USB-storage

Code

```
<<< <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
    <title></title>
    <script language="JavaScript" src="event.js"></script>
    <script>
      var stb = gSTB;
      var time = new Date();
//*****
*****
      var check_time = 12; //Set start adv time here (hour)
      var streamURL = 'udp://239.1.1.1:1234'; //Set channel (stream) url here
      var fileName = 'Peter Ilyich Tchaikovsky - Erich Kunzel - Cincinnati Pops
Orchestra - Tchaikovsky 1812 Overture (Hybrid SACD).flac'; //Set adv file
name here
//*****
*****
      var adv_timer = null;
      var adv_isPlaying = false;

      var stbEvent = {
        onEvent : function (event) {
          stb.Debug(event);
          if (event == 1 || event == 5) {
            stb.Play(streamURL);
          }
        },
        event : 0
      }

      function init() {
        stb.InitPlayer();
        time = new Date();
        if (time.getHours() == check_time) {
          eval('var usbs = ' + stb.RDir("get_storage_info"));
          stb.Stop();
          console.log(usbs[0].mountPath + '/' + fileName);
          stb.Play('ffmpeg ' + usbs[0].mountPath + '/' + fileName);
          adv_isPlaying = true;
        } else {
          stb.Play(streamURL);
        }
        adv_timer = window.setInterval(function () {
          time = new Date();
```

```
        if (time.getHours() == check_time) {
            eval('var usbs = ' + stb.RDir("get_storage_info"));
            stb.Stop();
            stb.Play('ffmpeg ' + usbs[0].mountPath + '/' + fileName);
            adv_isPlaying = true;
        }
    }, 3600000)
}
</script>
<body onload="init()" onKeyDown="getkeydown(event)">
</body>
</html>
```

Example of realization

Connect to STB via SSH.

login - root

pswd - 930920

Run the commands:

```
mv /home/web/services.html /home/web/services.html.old
vi /home/web/services.html
```

To copy the code of HTML page in services.html

Quit from “vi” editor with saving.

Reboot STB:

```
reboot
```

In this case HTML page <services.html> will be loaded with STB without main menu of the inner portal.

Mini Portal

Example 5.

Example of “mini portal” which will be loaded with STB run. It allows to play a certain number of streams with the ability to control STB with remote control. The user interface is a page with IPTV channels logo and ability to choose channel for watching..

[Download mini-portal](#)

Example "How to use"

Download the file. Unzip the file into your Web-server catalog. Those files must be available, for example: http://your_server/mini_portal/index.html

For the test run

To connect STB via SSH.

login: root

pswd: 930920

Run the commands:

```
killall stbapp
/usr/share/qt-4.6.0/stbapp -qws -display directfb
http://your_server/mini_portal/index.html
```

2D Navigation

Example 6.

Example of using 2D-navigation

Code

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="button/html; charset=UTF-8">

<title></title>

<script>
//gSTB.EnableSpatialNavigation(true);
var curelem=0,
    itemsAtLine=3;

function getkeydown(e){
    console.log();
    switch(e.keyCode || e.which){
        case 37:
            if(curelem){
                curelem--;
            }
        case 39:
            if(curelem < itemsAtLine-1){
                curelem++;
            }
        case 13:
            //gSTB.Click(curelem);
            curelem=0;
        default:
            return;
    }
}
```


DVB tuning / level of signal

Example 7.

Code

```
<html>

<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<script>
var curCh = 0;
var stbEvent=
{
    onEvent : function(data){},
    event : 0
}
var signalTm=0
function Info()
{
    gSTB.SetTopWin(0);
    signalTm = setInterval(UpdateSignalInfo,1000);
}
function UpdateSignalInfo()
{
    var level, quality, ber
    level = dvbManager.GetSignalLevel();
    quality = dvbManager.GetSignalQuality();
    document.getElementById('level').style.width = level + "%";
    document.getElementById('quality').style.width = quality + "%";
    ber = dvbManager.GetBitErrorRate();
    ber1 = ber/100000;
    if(ber1 > 100)
        ber1 = 100;
    document.getElementById('ber').style.width = ber1+"%";
    document.getElementById('ber').innerHTML = "BER: " + ber;
    gSTB.Debug("level: " + level + " quality: " + quality + " ber: " +
ber);
}
function keyDown(e)
{
    gSTB.Debug('which '+e.which);
    switch(e.which)
    {
        case 116: //refresh
            location=location;
            break;
    }
}
```

```
    }
}
function onTimer()
{

}
function OnTune()
{
gSTB.Debug("dvbManager.TuneChannel("+document.getElementById("TuneId").value
+",3)")
    dvbManager.TuneChannel(document.getElementById("TuneId").value,3);
    Info()
}
function OnUnTune()
{
    dvbManager.UnTuneChannel();
    clearInterval(signalTm);
}

</script>
<style type="text/css">
    body {
        background:#fff;
        padding:75px;
    }
    .bar{
        background-color:green;
        height:50px;
    }
</style>
</head>
<body bgcolor="beige"  onKeyPress="keyDown(event);"
onload="gSTB.InitPlayer();">
    <div id="debug"></div><div id="duration"></div>
    <div id="signal" style="position:absolute;
top:200px;left:200px;width:300px">
        <div>Level<div id="level" class="bar" ></div></div>
        <div>Quality<div id="quality" class="bar"></div></div>
        <div id="berH">BER<div id="ber" class="bar"></div></div>
    </div>
    <div id="scan"></div>
    <input id="TuneId" type="text" value="650000">
    <input type="button" value="Tune" onclick="OnTune()">
    <input type="button" value="UnTune" onclick="OnUnTune()">
</body>
</html>
```

From:

<http://docs.infomir.com.ua/> -

Permanent link:

http://docs.infomir.com.ua/doku.php?id=en:stb_webkit:faq:example_api

Last update: **2022/02/09 12:41**

