[Tutorial] LG Web_Handling Key Repeat Using JavaScript Events

Version 1.0 – October 2011

LGDEV-056

Home Entertainment Company
LG Electronics, Inc.

Copyright

About This Document

Revision History

Document Version	Date	Comment
1.0	October 12, 2011	Initial Version

Purpose

This document describes how JavaScript events are occurred when a key is pressed from remote control unit and to use these events in application.

Reference Documents

Refer to the following documents:
- LG Web Application Development Guide

Conventions

Codes

Source code and examples are indicated in the `grey Courier New` font.

Note, Caution

Note and caution are used to emphasize information.
The following samples describe when each is used.

Note

Contains information about something that is helpful to you.

Caution

Contains important information about something that you should know.

Abbreviation

The following table defines the abbreviations used in this document.

Abbreviation	Description
API	Application Programming Interface

Contents

Figures

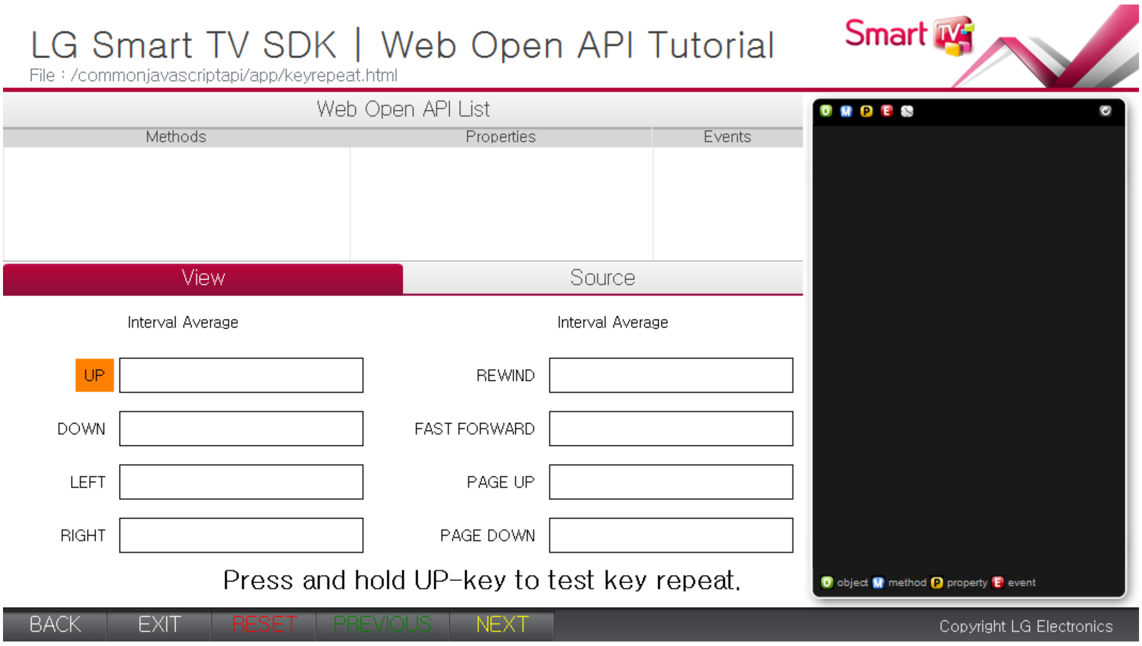
1 Introduction

This chapter provides an overview of this application.

1.1 Overview

1.1 Overview

This application shows how JavaScript events are occurred when a key is pressed from remote control unit and to use these events.



LG Smart TV SDK | Web Open API Tutorial

File : /commonjavascriptapi/app/keyrepeat.html

Press and hold UP-key to test key repeat.

[Figure 1] Key Repeat Application

Note

Log is used for checking the sequence of Web Open API; this will not be covered in the sample code description.

---

2 Creating Application

This chapter describes how JavaScript events are occurred when a key is pressed from remote control unit and to use these events.

2.1 Initializing the Page

Use the **initPage** function to set the basic functions of the application.

04: Record the last visited page when running the application.
07: Initialize the page.
08: Get the source code of the page using the XMLHttpRequest object.
09: Set the page ID.
10: Initialize the Log function.
13-16: Add an event handler which will executed when the corresponding button is pressed.
19-20: Add event handlers for key inputs (keydown, keyup) from remote control unit.
23: Set the background color of key to test.
24: Display description for test on screen.

Sample Code

```
01 : function initPage()
02 : {
03 :   //save page as last visited page
04 :   setLastVisitPage();
05 :
06 :   //common initialize function
07 :   commonInitialize();
08 :   requestSourceCode();
09 :   setPageID("Key Repeat");
10 :   jsLog.initLG();
11 :
12 :   //add onclick event handler
13 :   addEventHandler(document.getElementById("btn_back"),"click",onClickHandler);
14 :   addEventHandler(document.getElementById("btn_red"),"click", onClickHandler);
15 :   addEventHandler(document.getElementById("btn_green"),"click",onClickHandler);
16 :   addEventHandler(document.getElementById("btn_yellow"),"click",onClickHandler);
17 :
18 :   //add key repeat test event handler
19 :   addEventHandler(document.body, "keydown", onKeyDownHandler);
20 :   addEventHandler(document.body, "keyup", onKeyUpHandler);
21 :
22 :   //select current position
23 :   document.getElementById(keysToTest[currentTestPosition][0]).className =
     "eachTestGuide orangeBgColor";
24 :   setInnerTextById("keyrepeat_test_description", "Press and hold " +
     keysToTest[currentTestPosition][3] + "-key to test key repeat.");
25 : }
```

2.2 Handling Events

The following code shows event handler functions which are added from initPage() function.

02-16: Declare and initialize keyToTest, currentTestPosition, maxRepeatCount, maxGaugeWidth, repeatCount, startTime, and propressStatus.

onKeyDownHandler

This event handler called when key down is occurred.
20: Declare userInput and save key code of the event.
23: If userInput is same with setup key and the repeat number is not bigger than the maximum value, execute the following codes.
25-30: If repeatCount is 0, declare currentDate and initialize startTime to current time.
31: Increase repeatCount by 1.
32: Increase gauge bar by number of repeatCount.
34-40: If number of key repeat is bigger than maxRepeatCount, declare elapsedTime, count the key repeat time, and display it on screen.

onKeyUpHandler

This event handler is called when key up is occurred.
46: Declare userInput and save key code of the event.
49-52: If inputted key value is same with setup key value and the repeat number does not exceed the maximum number, change the color of corresponding gauge bar.

resetCurrentTest

Execute reset function.
57-59: Initialize repeatCount, startTime, and progressStatus.
60: Delete the child nodes of currentTestPosition.
61-63: Initialize the bar length of currentTestPosition and display the corresponding text on screen.

changeTestPosition

Change key to test.
68-70: Initialize repeatCount, startTime, and progressStatus.
71: Set class to unselect the selected key.
73-78: To go to previous key, decrease the value of currentTestPosition by 1 and move the position of the key.
If the value of currentTestPosition gets smaller than 0, move the position to the last key position according to the size of keysToTest.
80-85: To go to the next key, increase the value of currentTestPosition by 1 and move the key position.
If the value of currentTestPosition gets bigger than keysToTest, set to 0 and move to the first key position.
87-88: Changing the background color of class to orange to show that the testing key is changed. Also, display description text on screen.

Sample Code

```
01 : //[0] : title id, [1] : gauge id, [2] : key code, [3] : key name
02 : var keysToTest = [ ["up_title", "up_progress", 38, "UP"]
03 :                        , ["down_title", "down_progress", 40, "DOWN"]
04 :                        , ["left_title", "left_progress", 37, "LEFT"]
05 :                        , ["right_title", "right_progress", 39, "RIGHT"]
06 :                        , ["rewind_title", "rewind_progress", 412, "REWIND"]
07 :          ,["fast_forward_title" , "fast_forward_progress" , 417 , "FAST_FORWARD"]
08 :                        , ["page_up_title", "page_up_progress", 33, "PAGE_UP"]
09 :                        ,     ["page_down_title",     "page_down_progress",     34,
     "PAGE_DOWN"]];
10 :
11 : var currentTestPosition = 0;
12 : var maxRepeatCount = 40;
13 : var maxGaugeWidth = 273;
14 : var repeatCount = 0;
15 : var startTime = 0;
16 : var progressStatus = "";
17 :
18 : function onKeyDownHandler(event)
```

```
{
 var userInput = getKeyCode(event);
 jsLog.lgevent('javascript: Key Event Handling : KeyDown, Key : ' + getKeyName(event));

 if(userInput==keysToTest[currentTestPosition][2]&&repeatCount<maxRepeatCount)
 {
        if(repeatCount == 0)
        {
               var currentDate = new Date();
               startTime = currentDate.getTime();
               setInnerTextById(keysToTest[currentTestPosition][1], "");
        }
        repeatCount++;
        document.getElementById(keysToTest[currentTestPosition][1]).style.width = Math.floor((maxGaugeWidth * (repeatCount / maxRepeatCount))) + "px";

        if(repeatCount >= maxRepeatCount)
        {
               var currentDate = new Date();
               var elapsedTime = currentDate.getTime() - startTime;
               setInnerTextById(keysToTest[currentTestPosition][1], (elapsedTime / repeatCount) + "ms");
               setInnerTextById("keyrepeat_test_description", "Press YELLOW-Key to test next key.");
         }
 }
}

function onKeyUpHandler(event)
{
 var userInput = getKeyCode(event);
 jsLog.lgevent('javascript: Key Event Handling : KeyUp, Key : ' + getKeyName(event));

 if(userInput==keysToTest[currentTestPosition][2]&&repeatCount<maxRepeatCount)
 {
 document.getElementById(keysToTest[currentTestPosition][1]).style.backgroundColor="#FF0000";
 }
}

function resetCurrentTest()
{
 repeatCount = 0;
 startTime = 0;
 progressStatus = "";
 clearChildNodes(document.getElementById(keysToTest[currentTestPosition][1]));
 document.getElementById(keysToTest[currentTestPosition][1]).style.width="0px";
 document.getElementById(keysToTest[currentTestPosition][1]).style.backgroundColor = "#008000";
 setInnerTextById("keyrepeat_test_description", "Press and hold " + keysToTest[currentTestPosition][3] + "-key to test key repeat.");
}

function changeTestPosition(direction)
{
 repeatCount = 0;
 startTime = 0;
 progressStatus = "";
 document.getElementById(keysToTest[currentTestPosition][0]).className="eachTestGuide";

 if(direction == "PREVIOUS"){
        currentTestPosition--;
        if(currentTestPosition < 0){
               currentTestPosition = keysToTest.length - 1;
        }
 }

 if(direction == "NEXT"){
      currentTestPosition++;
      if(currentTestPosition >= keysToTest.length){
           currentTestPosition = 0;
      }
 }

 document.getElementById(keysToTest[currentTestPosition][0]).className = "eachTestGuide orangeBgColor";
 setInnerTextById("keyrepeat_test_description", "Press and hold " + keysToTest[currentTestPosition][3] + "-key to test key repeat.");
}
```

2.3 Source Code of keyrepeat.html

Source code of keyrepeat.html is as follows:

```
<!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=UTF-8">
<title>Key Repeat Test Page</title>
<link rel="stylesheet" href="../../css/style.css">
<script language="javascript" src="../../js/keycode.js"></script>
<script language="javascript" src="../../js/common.js"></script>
<script language="javascript" src="../../js/menu.js"></script>

<script type="text/javascript" src="../../blackbirdjs/blackbird.js"></script>
<link type="text/css" rel="Stylesheet" href="../../blackbirdjs/blackbird.css" />

<script>

	//initialize page
	function initPage()
	{
		//save page as last visited page
		setLastVisitPage();

		//common initialize function
		commonInitialize();
		requestSourceCode();
		setPageID("Key Repeat");
		jsLog.initLG();

		//add onclick event handler
		addEventHandler(document.getElementById("btn_back"), "click",
onClickHandler);
		addEventHandler(document.getElementById("btn_red"), "click",
onClickHandler);
		addEventHandler(document.getElementById("btn_green"), "click",
onClickHandler);
		addEventHandler(document.getElementById("btn_yellow"), "click",
onClickHandler);

		//add key repeat test event handler
		addEventHandler(document.body, "keydown", onKeyDownHandler);
		addEventHandler(document.body, "keyup", onKeyUpHandler);

		//select current position
		document.getElementById(keysToTest[currentTestPosition][0]).className =
"eachTestGuide orangeBgColor";
		setInnerTextById("keyrepeat_test_description", "Press and hold " +
keysToTest[currentTestPosition][3] + "-key to test key repeat.");
	}

	function onUserInput(userInput)
	{
		switch(userInput)
		{
			case VK_BACK :
window.location.replace("../menu_commonjavascriptapi.html"); break;
			case VK_RED : case 82 : resetCurrentTest(); break;
			case VK_GREEN : case 71 : changeTestPosition("PREVIOUS"); break;
			case VK_YELLOW : case 89 : changeTestPosition("NEXT"); break;
		}
	}

	//array of keys to test
	//[0] : title id, [1] : gauge id, [2] : key code, [3] : key name
	var keysToTest = [ ["up_title", "up_progress", 38, "UP"]
				  , ["down_title", "down_progress", 40, "DOWN"]
```

```
                                 , ["left_title", "left_progress", 37, "LEFT"]
                                 , ["right_title", "right_progress", 39, "RIGHT"]
                                 , ["rewind_title", "rewind_progress", 412, "REWIND"]
                                 , ["fast_forward_title", "fast_forward_progress",
417, "FAST_FORWARD"]
                                 , ["page_up_title", "page_up_progress", 33,
"PAGE_UP"]
                                 , ["page_down_title", "page_down_progress", 34,
"PAGE_DOWN"]];

      var currentTestPosition = 0;
      var maxRepeatCount = 40;
      var maxGaugeWidth = 273;
      var repeatCount = 0;
      var startTime = 0;
      var progressStatus = "";

      function onKeyDownHandler(event)
      {
             var userInput = getKeyCode(event);
             jsLog.lgevent('javascript: Key Event Handling : KeyDown, Key : ' +
getKeyName(event));

             if(userInput == keysToTest[currentTestPosition][2] && repeatCount <
maxRepeatCount){
                   if(repeatCount == 0){
                         var currentDate = new Date();
                         startTime = currentDate.getTime();
                         setInnerTextById(keysToTest[currentTestPosition][1], "");
                   }
                   repeatCount++;

      document.getElementById(keysToTest[currentTestPosition][1]).style.width =
Math.floor((maxGaugeWidth * (repeatCount / maxRepeatCount))) + "px";
                   if(repeatCount >= maxRepeatCount){
                                var currentDate = new Date();
                                var elapsedTime = currentDate.getTime() - startTime;
                                setInnerTextById(keysToTest[currentTestPosition][1],
(elapsedTime / repeatCount) + "ms");
                         setInnerTextById("keyrepeat_test_description", "Press YELLOW-
Key to test next key.");
                    }
             }
      }

      function onKeyUpHandler(event)
      {
             var userInput = getKeyCode(event);
             jsLog.lgevent('javascript: Key Event Handling : KeyUp, Key : ' +
getKeyName(event));

             if(userInput == keysToTest[currentTestPosition][2] && repeatCount <
maxRepeatCount){

      document.getElementById(keysToTest[currentTestPosition][1]).style.backgroundColor
= "#FF0000";
             }
      }
      //reset current key test status
      function resetCurrentTest()
      {
             repeatCount = 0;
             startTime = 0;
             progressStatus = "";

      clearChildNodes(document.getElementById(keysToTest[currentTestPosition][1]));
             document.getElementById(keysToTest[currentTestPosition][1]).style.width =
"0px";

      document.getElementById(keysToTest[currentTestPosition][1]).style.backgroundColor
= "#008000";
```

```
                setInnerTextById("keyrepeat_test_description", "Press and hold " +
keysToTest[currentTestPosition][3] + "-key to test key repeat.");
        }
        //change test position
        function changeTestPosition(direction)
        {
                repeatCount = 0;
                startTime = 0;
                progressStatus = "";
                document.getElementById(keysToTest[currentTestPosition][0]).className =
"eachTestGuide";

                if(direction == "PREVIOUS"){
                        currentTestPosition--;
                        if(currentTestPosition < 0){
                                currentTestPosition = keysToTest.length - 1;
                        }
                }
                if(direction == "NEXT"){
                currentTestPosition++;
                if(currentTestPosition >= keysToTest.length){
                        currentTestPosition = 0;
                        }
                }
                document.getElementById(keysToTest[currentTestPosition][0]).className =
"eachTestGuide orangeBgColor";
                setInnerTextById("keyrepeat_test_description", "Press and hold " +
keysToTest[currentTestPosition][3] + "-key to test key repeat.");
        }

        function getKeyName(event)
        {
                var key = getKeyCode(event);
                for(var i = 0 ; i < keysToTest.length ; i++)
                {
                        if(keysToTest[i][2] == key)
                        {
                                return keysToTest[i][3];
                        }
                }

                return "-";
        }

</script>
</head>
<body ondragstart='return false' onselectstart='return false'
onload="javascript:initPage();">

<!-- title -->
<div class='SuiteTitle' >LG Smart TV SDK | Web Open API Tutorial</div>

<!-- navigation -->
<div class='SuiteNavigation'>
        <div style="float:left;">File : /commonjavascriptapi/app/keyrepeat.html</div>
</div>
<div class='SuiteTitleLine'> </div>

<!-- test contents -->
<div id='ContentArea' class='ContentArea' >
        <div class='ApiListTitleArea'>Web Open API List</div>
        <div class='ApiListArea'>
                <div class='MethodTitleArea'>
                        Methods
                        <div class='MethodListArea'>

                        </div>
                </div>
                <div class='PropertyTitleArea'>
                        Properties
                        <div class='PropertyListArea'>
                        </div>
                </div>
                <div class='EventTitleArea'>
                        Events
                        <div class='EventListArea'>

                        </div>
                </div>
        </div>
        <div class='ViewTitleArea'>
                <div id='tabViewArea' class='SelectedViewArea' style='float:left;'
onclick="showView();">View</div>
                <div id='tabCodeArea' class='UnselectedViewArea' style='float:right;'
onclick='showCode();'>Source</div>
        </div>
        <div id='view'>
        <div class = 'ViewArea'>
                <table width="880px" border="0" cellpadding="0" cellspacing="0">
                <tr height="60px">
                        <td width="150px"></td>
                        <td width="275px"><div class='eachTestGuide'>Interval
Average</div></td>
                        <td width="275px"></td>
                        <td width="275px"><div class='eachTestGuide'>Interval
Average</div></td>
                </tr>
                <tr height="60px">
                        <td width="150px" align="right"><div class='eachTestGuide'
id="up_title">UP</div>&nbsp;</td>
                        <td width="275px" align="left"><div class='gaugeBorder'><div
class='gaugeBar' id="up_progress"></div></div></td>
                        <td width="275px" align="right"><div class='eachTestGuide'
id="rewind_title">REWIND</div>&nbsp;</td>
                        <td width="275px" align="left"><div class='gaugeBorder'><div
class='gaugeBar' id="rewind_progress"></div></div></td>
                </tr>
                <tr height="60px">
                        <td width="150px" align="right"><div class='eachTestGuide'
id="down_title">DOWN</div>&nbsp;</td>
                        <td width="275px" align="left"><div class='gaugeBorder'><div
class='gaugeBar' id="down_progress"></div></div></td>
                        <td width="275px" align="right"><div class='eachTestGuide'
id="fast_forward_title">FAST FORWARD</div>&nbsp;</td>
                        <td width="275px" align="left"><div class='gaugeBorder'><div
class='gaugeBar' id="fast_forward_progress"></div></div></td>
                </tr>
                <tr height="60px">
                        <td width="150px" align="right"><div class='eachTestGuide'
id="left_title">LEFT</div>&nbsp;</td>
                        <td width="275px" align="left"><div class='gaugeBorder'><div
class='gaugeBar' id="left_progress"></div></div></td>
                        <td width="275px" align="right"><div class='eachTestGuide'
id="page_up_title">PAGE UP</div>&nbsp;</td>
                        <td width="275px" align="left"><div class='gaugeBorder'><div
class='gaugeBar' id="page_up_progress"></div></div></td>
                </tr>
                <tr height="60px">
                        <td width="150px" align="right"><div class='eachTestGuide'
id="right_title">RIGHT</div>&nbsp;</td>
                        <td width="275px" align="left"><div class='gaugeBorder'><div
class='gaugeBar' id="right_progress"></div></div></td>
                        <td width="275px" align="right"><div class='eachTestGuide'
id="page_down_title">PAGE DOWN</div>&nbsp;</td>
                        <td width="275px" align="left"><div class='gaugeBorder'><div
class='gaugeBar' id="page_down_progress"></div></div></td>
                </tr>
                </table>
        </div>
        </div>
        <div    scrolling="AUTO" style="visibility: hidden" id='codeview'>
```

```
		<textarea class="SourceCodeArea" value=""  id='sourcecode'></textarea>
	</div>

</div>

<!-- description -->
<div id='keyrepeat_test_description' class='SuiteDescription' style="position:
absolute; left: 40px; top:630px"></div>

<!-- button and copyright -->
<div class='SuiteButtonArea'>
	<!-- back key description -->
	<div id='btn_back' class='buttonDescription '>BACK</div>

	<!-- exit key description -->
	<div id='btn_exit' class='buttonDescription'>EXIT</div>

	<!-- red key description -->
	<div id='btn_red' class='buttonDescription redColor'>RESET</div>

	<!-- green key description -->
	<div id='btn_green' class='buttonDescription greenColor'>PREVIOUS</div>

	<!-- yellow key description -->
	<div id='btn_yellow' class='buttonDescription yellowColor'>NEXT</div>

	<!-- copyright -->
	<div class='copyright'>Copyright LG Electronics</div>
</div>

</body>
</html>
```