

Virtual Keyboard

Virtual Keyboard is designed to appear as below when a text input element is focused. Developers can easily use virtual keyboard by including virtual keyboard library.

USB keyboard input and Voice Recognition are supported since NetCast 3.0(2012).

To download the Virtual Keyboard Library, go to **[Resource Center > Smart TV > SDK & Tools > Sample Codes]** menu in LG Developer (<http://developer.lge.com>) website.



Using Virtual Keyboard Library

To use the virtual keyboard, add the following script in the head of html file. In order to use virtual keyboard in iframes, the additional script should be added in the html file connected to the iframe. (Relative path of the jsLgVKeyboard folder should be modified according to the real path.)

In a normal or parent page

```
<link type="text/css" rel="Stylesheet" href="../jsLgVKeyboard/LgVKeyboard.css" />
<script id="mainVKScript" type="text/javascript" src="../jsLgVKeyboard/LgVKeyboard.js">
</script>
```

In iframe pages

```
<script type="text/javascript" src="../jsLgVKeyboard/LgVKeyboardIframe.js"></script>
```

Virtual Keyboard Object(IgKb) uses these Keyboard APIs.

For the 2014 model year LG Smart TVs, they are designed to use the System Keyboard not creating the Virtual Keyboard Object (IgKb) even when using the Virtual Keyboard Library. In this regard, please check for sure if the IgKb object has been created first and then use the Keyboard API.

setInitEventHandler

Description

Sets the event handler that is going to be called when Virtual Keyboard is completely initialized.

Parameter

initEventHandler Event handler that receives the "Initialization Complete" event.

Return Value

None

isInitialized

Description

Notifies if Virtual Keyboard is initialized or not.

Parameter

None

Return Value

initStatus : Initialization is complete or not. [boolean]

isShown

Description

Notifies if Virtual Keyboard is displayed or not. This function is available only after Virtual Keyboard is completely initialized.

Parameter

None

Return Value

show : Virtual Keyboard is displayed or not. [boolean]

onKeyDown

Description

Event Handler that is occurred when Remote Control Unit or Key of HID Keyboard is down. Developers should use IgKb.onKeyDown instead of window.onkeydown as the Virtual Keyboard uses the keydown event. This is able to use after the initialization of Virtual Keyboard is completed.

Parameter

event : event data of window.onkeydown

Return Value

None

onKeyUp

Description

Event Handler that is occurred when Remote Control Unit or Key of HID Keyboard is up. Developers should use IgKb.onKeyUp instead of window.onkeyup as the Virtual Keyboard uses the keyup event. This is able to use after the initialization of Virtual Keyboard is completed.

Parameter

event : event data of window.onkeyup

Return Value

None

onShow

Description

Event Handler that is occurred when Virtual Keyboard is shown. This is able to use after the initialization of Virtual Keyboard is completed.

Parameter

None

Return Value

None

onHide

Description

Event Handler that is occurred when Virtual Keyboard is hidden. This is able to use after the initialization of Virtual Keyboard is completed.

Parameter

None

Return Value

None

targetBgColor

Description

Background color of the element that is going to receive the key input of Virtual Keyboard. This is able to use after the initialization of Virtual Keyboard is completed.

targetTextColor

Description

Text color of the element that is going to receive the key input of Virtual Keyboard. This is able to use after the initialization of Virtual Keyboard is completed.

- Do not modify the folder structure and file name of jsLgVKeyboard.
- The script must include the following declaration: **id = "mainVKScript"**
- A web page should be set character set as UTF-8 to use virtual keyboard as following:

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

- Virtual keyboard is enabled in the following tag/types:

| Tag | Type |
|----------|----------|
| INPUT | TEXT |
| INPUT | PASSWORD |
| TEXTAREA | |

- Beware of cross-domain problem when using iframe. If an application uses multiple html files those are loaded from different domains, cross-domain problem occurs and virtual keyboard cannot be used in iframe.

Supported Language

Virtual keyboard supports 42 languages, and 16 languages are supported for voice recognition of them. Refer to **Developing > API > Voice Recognition Plugin and API** section in this Library for detailed information.

Sample Codes

```
<!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>Insert title here</title>
```

```
<link type="text/css" rel="Stylesheet" href="./jsLgVKeyboard/LgVKeyboard.css" />
<script id="mainVKScript" type="text/javascript"
src="./jsLgVKeyboard/LgVKeyboard.js"></script>

<script>
function onKeyDown(event)
{
    var keyCode = event.keyCode;

    switch(keyCode) {
        case VK_LEFT :
            document.getElementById('text1.3').blur();
            document.getElementById('text1.1').focus();
            break;

        case VK_RIGHT :
            document.getElementById('text1.1').blur();
            document.getElementById('text1.3').focus();
            break;
    }

    event.returnValue = false;
}

function addInputBox()
{
    document.getElementById("addInput").innerHTML = '<input type="text" id="text1.4"
value="test 1.4" style="width:180px" role="textbox" style="color: yellow; background-color:
gray">';
}

function updateVKStatus()
{
    document.getElementById("output").value = "Shown? : " + lgKb.isShown();
}

function onInitialized()
{
    setInterval(updateVKStatus, 3000);
    lgKb.onShow = function () { document.getElementById("output").value = "Show event
occur"; };
    lgKb.onHide = function () { document.getElementById("output").value = "Hide event
occur"; };

    lgKb.onKeyDown = onKeyDown;
    lgKb.targetBgColor = "#934910";
    lgKb.targetTextColor = "blue";

    document.getElementById('text1.1').focus();
}

function initPage()
{
    if(window.lgKb == null) // for 2014 TV
    {
        document.getElementById('text1.1').focus();
    }
}
</script>
```

```
    }
    else
    {
        lgKb.setInitEventHandler(onInitialized);
    }
}
</script>

</head>
<body onload="initPage();">
    <div>Lg Virtual Keyboard Test</div>
    <div style="height: 150px;">
        <input type="text" id="text1.1" value="test 1.1" style="width:180px" role="textbox"
style="color: yellow; background-color: gray">
        <input type="password" id="text1.2" value="test 1.2" style="width:180px">
        <textarea rows="4" cols="20" id="text1.3" value="test 1.3"> </textarea>
        <input type="text" id="output" style="width:180px" style="color: yellow; background-
color: gray">
        <div id="addInput"> </div>
        <input type="button" value="refresh" onclick="location.reload();">
        <input type="button" value="Add" onclick="addInputBox();">
    </div>
</body>
</html>
```