LG Electronics

Web App Development Quick Start Guide (WebAppSample_ImageViewer)

Version 1.0.0 – January 2013

LGDEV-089

Home Entertainment Company
LG Electronics, Inc.

Copyright

About This Document

Revision History

Document Version	Date	Comment
1.0.0	2013, January 10	Initial Release

Purpose

This document is a quick guide intended to describe how developers develop and test web applications using the WebAppSample_ImageViewer example. This will help developer to understand how to use Image Viewer Framework APIs in the application.

Reference Documents

Refer to the following documents:
- Developer Library in LG Developer (**http://developer.lge.com**) website

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
CSS	Cascading Style Sheet
SDK	Software Development Kit

Contents

1 Introduction 5

1.1 Overview of WebAppSample_ImageViewer 6

1.2 Structure and Resources of Sample Project 7

2 Setting Up the Project 8

3 Implementing the Code 9

3.1 Descriptions on Source Codes 10

3.2 Expected Result 17

4 Testig the Application 20

Figures

[Figure 1] WebAppSample_ImageViewer 6

[Figure 2] Structure and Resources of WebAppSample_ImageViewer Project 7

Tables

[Table 1] Structure of WebAppSample_ImageViewer Project 7

1 Introduction

This chapter describes WebAppSample_ImageViewer and its structure.

1.1 Overview of WebAppSample_ImageViewer
1.2 Structure and Resources of Sample Project

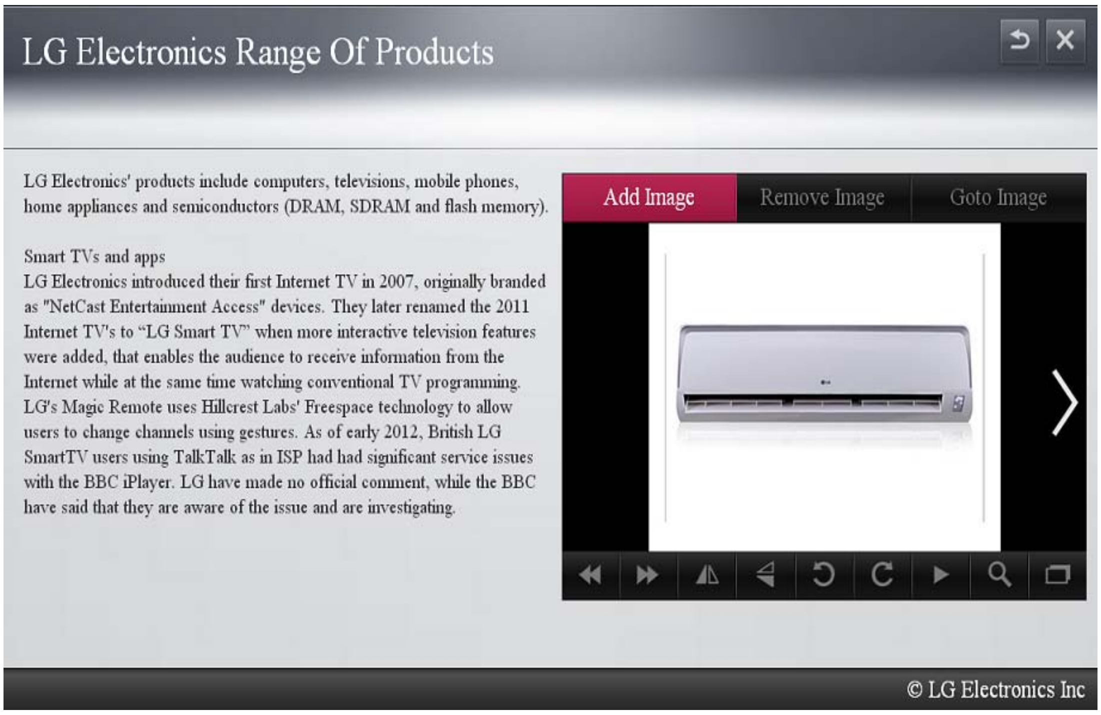
1.1 Overview of WebAppSample_ImageViewer

A web application that is developed based on general HTML standards has the following structure for its operation. If you separate each component independently, it is easy to manage application sources. Developers need to implement each component as necessary.

- HTML: Defines application structure
- JavaScript: Defines application operation
- CSS: Defines application design elements

In this guide, we are going to learn how to develop and test a web application using a sample development example that uses Image Viewer Framework in the application. Developer can use existing control of framework or create their own controls and use the API exposed by framework.

The operation of WebAppSample_ImageViewer application is shown below. The controls such as 'Add Image', 'Remove Image', 'Goto Image', 'Back', 'Exit' and so on are provided to control the application. Developer can perform basic operation like zoom, flip, rotate and so on by simply clicking on control button. It will work both in normal and full screen display mode.



LG Electronics' products include computers, televisions, mobile phones, home appliances and semiconductors (DRAM, SDRAM and flash memory).

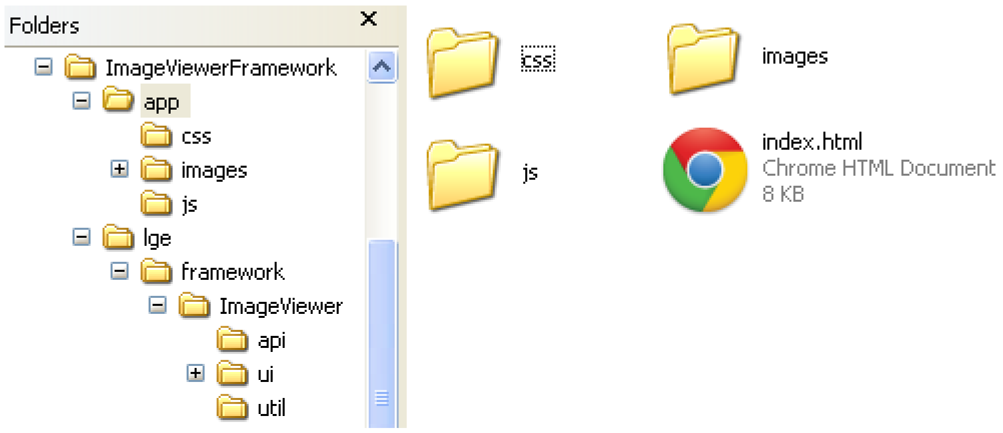
Smart TVs and apps
LG Electronics introduced their first Internet TV in 2007, originally branded as "NetCast Entertainment Access" devices. They later renamed the 2011 Internet TV's to "LG Smart TV" when more interactive television features were added, that enables the audience to receive information from the Internet while at the same time watching conventional TV programming. LG's Magic Remote uses Hillcrest Labs' Freespace technology to allow users to change channels using gestures. As of early 2012, British LG SmartTV users using TalkTalk as in ISP had had significant service issues with the BBC iPlayer. LG have made no official comment, while the BBC have said that they are aware of the issue and are investigating.

[Figure 1] WebAppSample_ImageViewer

1.2 Structure and Resources of Sample Project

The file structure of WebAppSample_ImageViewer application that is provided as a sample is shown below. You can download the sample codes from [Resource Center > Smart TV > SDK & Tools > Tools & Samples] menu in LG Developer (**http://developer.lge.com**) website.

You can either copy this sample source into the WebContent folder of the project that is created in LG IDE or enter the code directly by referring to the description in this guide. This sample project uses the Image viewer framework. Therefore, developer must include lge folder into the WebContent folder also.



[Figure 2] Structure and Resources of WebAppSample_ImageViewer Project

[Table 1] Structure of WebAppSample_ImageViewer Project

Folder		File	Description
Image Viewer Sample		index.html	Initial operation file of an application
	css	style.css	Design elements of an application
	js	main.js	JavaScript files to operate an application
	Image		Image files of an application
Image Viewer Framework	api	api.js	JavaScript to operate on framework
	util	keycode.js	Key code definitions and value assignments
	ui > css	ui.css	Design elements of framework
	ui > Images		Image files of a framework

2 Setting Up the Project

Although it is okay to use a general editor to develop a web application, you can easily develop, debug, and test the application if you use the LG IDE. This chapter describes how to set up a project using the LG IDE.

Note

For detailed information on how to start using LG IDE, refer to **Developing > Using SDK** section in Developer Library in LG Developer (**http://developer.lge.com**) website.

1. Open LG IDE and click **[File > New > LG Web Project]**.

2. In the 'New LG Web Project' window, give a project name and select 'SDK Version for Web Application'. Then, click **[Finish]**.

3. Add JavaScript, CSS, and image files.

Create each folder for JavaScript, CSS, and image files that will be used in this example in the WebContent folder. Although folders are not mandatory, it will be helpful to manage resources using folders when considering resource increase in the future.

Select and right-click the WebContent node and select **[File]** and then enter a file name including its extension in the file name field to add JavaScript or a CSS file.

If you want to use any external library for your application, add library folder (in this case lge folder).

4. Now, a basic foundation is prepared to implement a sample example.

3 Implementing the Code

This chapter describes the source codes, description, and expected results of WebAppSample_ImageViewer example.

3.1 Descriptions on Source Codes
3.2 Expected Result

Note

For detailed information on how to use the image viewer framework APIs, refer to **Developing > API** section in Developer Library in LG Developer (**http://developer.lge.com**) website..

3.1 Descriptions on Source Codes

This section describes the core part of the sample source codes.
Following sources are related to index.html, which is for Main Screen.

Linking JavaScript and CSS (index.html)

Specifies the path of JavaScript, stylesheet and library(if any) source file that will be used in an application. As soon as document is loaded, call init() method.

```
<link rel="stylesheet" type="text/css" href="css/style.css" />
<script language="javascript" type="text/javascript" src="js/jquery-
1.8.2.min.js"></script>
<script language="javascript" type="text/javascript"
src="../lge/framework/ImageViewer/api/api.js"></script>
<script language="javascript" type="text/javascript"
src="../lge/framework/ImageViewer/util/keycode.js"></script>
<script language="javascript" type="text/javascript" src="js/main.js"></script>
<script>
         $(document).ready(function (e) {
             init();
         });
</script>
```

Designing displaying structure (index.html)

Designs the structure to be displayed on an application screen using the <div> tag.

```
<div id="wraper">
wraper area

<div class="header">
Top area

<p class="title">LG Electronics Range Of Products</p>
Title area

<ul class="main_btn">
The area for back or exit button

<li id="returnBtn" class="btn_default"></li>
Return Button
<li id="closeBtn" class="btn_default"></li>
Close button

</ul>
</div>


<div class="container">
Container Area

<div class=" contents">
Area for text Contents
```

```
</div>

<div class="nomal_imageviewer">
Image Viewer Area


<ul class="top_btn btn01">
Sample Controls Area


<li id="addImgBtn" class="btn_default"> </li>
Add Image Button
<li id="removeImgBtn" class="btn_default"> </li>
Remove Image Button
<li id="goToImgBtn" class="btn_default btn_last"> </li>
GoTo Image Button


</ul>

<div id="myGallery" class="myGallery" style="width:610px; height:384px">
Image Viewer Framework Area with width and height
<ul>
     <li>
        <img src="images/airCondition.jpg">
          List of Images to be included for image viewer
     </li>
</ul>

</div>

</div>

</div>


<div class=" footer">

Bottom area

</div>

</div>
```

Implementing function for document ready (main.js)

Implements a JavaScript code which will execute on document ready event.

```
01 : function init(){
02 :    app = new lge();
03 :    app.createImgViewer($("#myGallery"), true, clearMainFocus);
04 :    bindEvents();
05 :    $('#myGallery').addClass("myGalleryFocus");
06 :    appFocussedCtrl = "myGallery";
07 :    var elem = document.getElementById("myGallery");
08 :    elem.addEventListener("mousewheel", app.mouseWheel, false);
09 : }
```

02: Creating the object of lge class.
03: Calling createImgViewer method of lge class with following three parameter.
  - Id of framework container div
  - Boolean variable true for including framework control and its key navigation
  - Callback method for removing application focus from framework
04: Calling bindEvents method for binding mouse and click events of controls.
05: Adding focus to framework container.
06: Setting the myGallery as current focused object by storing it into appFocussedCtrl variable.
07: Getting the id of framework container i.e. myGallery.
08: Adding the mousewheel event listener to myGallery and calling mouseWheel API for changing

the image on mouse role.

Binding key and mouse event to controls (main.js)

Implements a JavaScript code which will execute on mouse enter, key down and click event.

Calling handleKeyDown(key) method for key down event on document, handleMouseEnter() for mouse enter and handleClick() method for Click event for all buttons and myGallery container.

```
function bindEvents() {
    $(document).keydown(function (event) {
        var key = event.keycode || event.which;

        handleKeyDown(key);

    });
    $('#closeBtn').mouseenter(function (e) {

        handleMouseEnter(this);

    });
    $('#closeBtn').click(function (e) {

        handleClick(this);

    });

    $('#returnBtn').mouseenter(function (e) {

        handleMouseEnter(this);

    });
    $('#returnBtn').click(function (e) {

        handleClick(this);

    });

    $('#addImgBtn').mouseenter(function (e) {

        handleMouseEnter(this);

    });
    $('#addImgBtn').click(function (e) {

        handleClick(this);

    });

    $('#removeImgBtn').mouseenter(function (e) {

        handleMouseEnter(this);

    });
    $('#removeImgBtn').click(function (e) {

        handleClick(this);

    });

    $('#goToImgBtn').mouseenter(function (e) {

        handleMouseEnter(this);

    });
    $('#goToImgBtn').click(function (e) {

        handleClick(this);

    });

    $('#myGallery').mouseenter(function (e) {
        handleMouseEnter(this);

    });
}
```

Processing mouse enter events(main.js)

Implements a JavaScript code about mouse enter event.

Calling the clearPreviousFocus() method for removing the focus from all other focused element.

If mouse is entered on myGallery then add focus to it and call setFrameworkFocus API for adding focus to appropriate element of framework based on mouse position on the framework.

If mouse is entered on any of the application control button then first remove focus from myGallery and add focus to respective button of the application.

```
function handleMouseEnter(element) {
    clearPreviousFocus();
    appFocussedCtrl = $(element).attr("id");

    switch (appFocussedCtrl) {
        case "myGallery":

            $(element).addClass("myGalleryFocus");

            app.setFrameworkFocus("down");
            break;
        case "closeBtn":
        case "returnBtn":
        case "addImgBtn":
        case "removeImgBtn":
        case "goToImgBtn":

            $("#myGallery").removeClass("myGalleryFocus");
            $(element).removeClass("btn_default").addClass("btn_focus");
            break;

    }
}
```

Implementing actions for button click(main.js)

Implements a JavaScript code for performing appropriate operation when you click on a button in the screen.

doSelection API is used to handle the click event of framework controls. On click of framework control it will call corresponding API.

```
case "myGallery":
            app.doSelection();

            break;
```

NetCastExit method is used to exit or quit the application to AV.

```
case "closeBtn":
            if (window.NetCastExit) {
                window.NetCastExit();

            }
```

NetCastBack method is used to return to my app screen on click of return button.

```
case "returnBtn":
            if (window.NetCastBack) {
                window.NetCastBack();
            }
```

addImage API is used to add image provided by path url as parameter in the last of image list.
removeFrameworkFocus API is used to remove focus from framework controls.

```
case "addImgBtn":
            app.addImage("images/television.jpg");
            $("#myGallery").removeClass("myGalleryFocus");
            app.removeFrameworkFocus();
```

removeImage API is used to remove current image from the list. If there is only one image in the list, it will not remove image from the list.
removeFrameworkFocus API is used to remove focus from framework controls.

```
case "removeImgBtn":
            app.removeImage();
            $("#myGallery").removeClass("myGalleryFocus");
            app.removeFrameworkFocus();
```

gotoImage API is used to display the required index of image from image list. If invalid index number is passed, it will show toast to user.
removeFrameworkFocus API is used to remove focus from framework controls.

```
case "goToImgBtn":
            app.gotoImage(2);
            $("#myGallery").removeClass("myGalleryFocus");
            app.removeFrameworkFocus();
```

Processing key event (main.js)

Defines action according to the remote controller key input.

On press of right key of remote control:

If focus is on framework, focusNext() API is used to remove focus to next control of framework.
If focus is on application control, it will add focus to next control of application.

```
case VK_RIGHT:
            if ($("#" + appFocussedCtrl).hasClass("myGalleryFocus")) {
                app.focusNext();
             } else {
                if (appFocussedCtrl == "removeImgBtn") {
                    clearPreviousFocus();
                    appFocussedCtrl = "goToImgBtn";
                    $("#" +
appFocussedCtrl).removeClass("btn_default").addClass("btn_focus");


                }
```

On press of left key of remote control:

If focus is on framework, focusPrevious() API is used to remove focus to previous control of framework.
If focus is on application control, it will add focus to previous control of application.

```
case VK_LEFT:
```

```
            if ($("#" + appFocussedCtrl).hasClass("myGalleryFocus")) {
                app.focusPrevious();
            } else {
                if (appFocussedCtrl == "goToImgBtn") {
                    clearPreviousFocus();
                    appFocussedCtrl = "removeImgBtn";
                    $("#" +
appFocussedCtrl).removeClass("btn_default").addClass("btn_focus");
              }
```

On press of up key of remote control:

If focus is on framework, focusUp() API is used to move focus to one step up control. In case of top control of framework, this API will return false, then focus will move to application controls.
If focus is on application control, it will add focus to one step up control of application.

```
case VK_UP:
            if ($("#" + appFocussedCtrl).hasClass("myGalleryFocus")) {
                if (app.focusUp() == false && app.getDisplayMode() == "Normal") {
                    clearPreviousFocus();
                    appFocussedCtrl = "addImgBtn";
                    $("#" +
appFocussedCtrl).removeClass("btn_default").addClass("btn_focus");
                }
            } else if (appFocussedCtrl == "addImgBtn" || appFocussedCtrl ==
"removeImgBtn" || appFocussedCtrl == "goToImgBtn") {

                clearPreviousFocus();
                appFocussedCtrl = "returnBtn";
                $("#" +
appFocussedCtrl).removeClass("btn_default").addClass("btn_focus");

            }         app.removeFrameworkFocus();
```

On press of down key of remote control:

If focus is on framework, focusDown() API is used to move focus to one step down control of framework. If focus is on last line control then this API will return false.
If focus is on application control, it will add focus to one step down control of application.

```
case VK_DOWN:
            if ($("#" + appFocussedCtrl).hasClass("myGalleryFocus")) {
                app.focusDown();
            } else {
           if (appFocussedCtrl == "returnBtn" || appFocussedCtrl == "closeBtn") {
                    clearPreviousFocus();
                    appFocussedCtrl = "addImgBtn";
                    $("#" +
appFocussedCtrl).removeClass("btn_default").addClass("btn_focus");
                   }
```

On press of OK key of remote control it will call doSelection() method.

```
case VK_ENTER:
            doSelection();
```

Removing focus from Control (main.js)

removeFrameworkFocus API is used to remove focus from framework controls.
If focus is on application control, it will remove focus from application control.

```
function clearPreviousFocus() {
    switch (appFocussedCtrl) {
        case "myGallery":
            app.removeFrameworkFocus();
            break;
        case "closeBtn":
        case "returnBtn":
        case "addImgBtn":
        case "removeImgBtn":
        case "goToImgBtn":
            $("#" +
appFocussedCtrl).removeClass("btn_focus").addClass("btn_default");
            break;
    }

}
```

Removing Focus from application control (main.js)

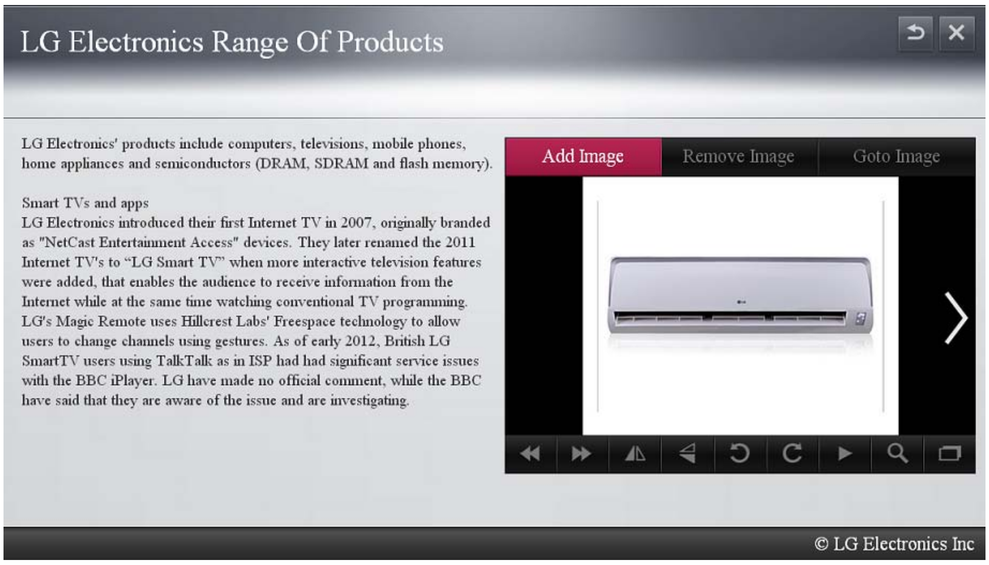
This method will remove focus from application controls and add focus to framework. It is used as callback function in framework for clearing application focus.

```
function clearMainFocus() {
    $("#" + appFocussedCtrl).removeClass("btn_focus").addClass("btn_default");
    appFocussedCtrl = "myGallery";
    $("#myGallery").addClass("myGalleryFocus");
}
```

3.2 Expected Result

The following screens will be displayed when the sample application starts on the LG Smart TV Emulator 2012. When the application starts, the initial screen is displayed. When a user clicks a button, corresponding operation will start.

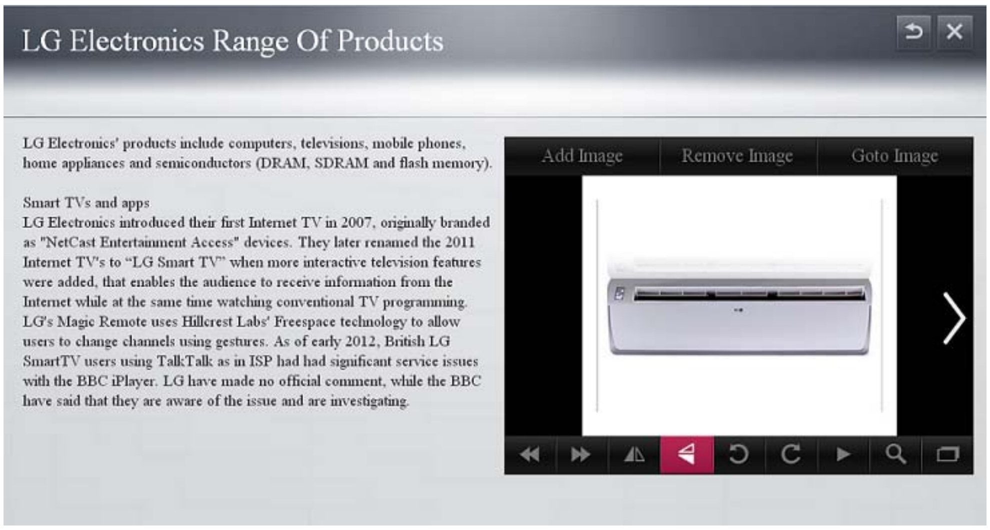
1) Initial Screen



LG Electronics' products include computers, televisions, mobile phones, home appliances and semiconductors (DRAM, SDRAM and flash memory).

Smart TVs and apps
LG Electronics introduced their first Internet TV in 2007, originally branded as "NetCast Entertainment Access" devices. They later renamed the 2011 Internet TV's to "LG Smart TV" when more interactive television features were added, that enables the audience to receive information from the Internet while at the same time watching conventional TV programming. LG's Magic Remote uses Hillcrest Labs' Freespace technology to allow users to change channels using gestures. As of early 2012, British LG SmartTV users using TalkTalk as in ISP had had significant service issues with the BBC iPlayer. LG have made no official comment, while the BBC have said that they are aware of the issue and are investigating.

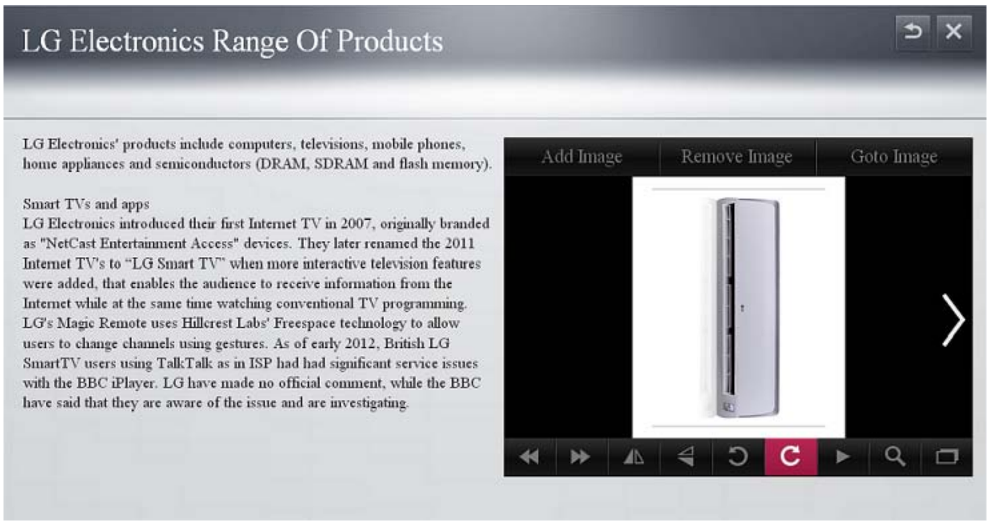
2) After pressing flip vertical button



LG Electronics' products include computers, televisions, mobile phones, home appliances and semiconductors (DRAM, SDRAM and flash memory).

Smart TVs and apps
LG Electronics introduced their first Internet TV in 2007, originally branded as "NetCast Entertainment Access" devices. They later renamed the 2011 Internet TV's to "LG Smart TV" when more interactive television features were added, that enables the audience to receive information from the Internet while at the same time watching conventional TV programming. LG's Magic Remote uses Hillcrest Labs' Freespace technology to allow users to change channels using gestures. As of early 2012, British LG SmartTV users using TalkTalk as in ISP had had significant service issues with the BBC iPlayer. LG have made no official comment, while the BBC have said that they are aware of the issue and are investigating.

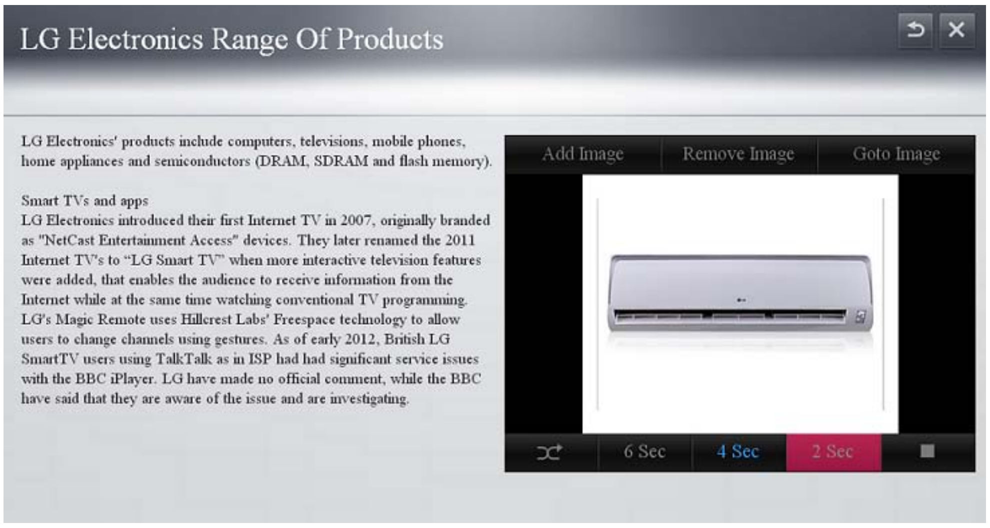
3) After pressing rotate right info button


LG Electronics Range Of Products

LG Electronics' products include computers, televisions, mobile phones, home appliances and semiconductors (DRAM, SDRAM and flash memory).

Smart TVs and apps
LG Electronics introduced their first Internet TV in 2007, originally branded as "NetCast Entertainment Access" devices. They later renamed the 2011 Internet TV's to "LG Smart TV" when more interactive television features were added, that enables the audience to receive information from the Internet while at the same time watching conventional TV programming. LG's Magic Remote uses Hillcrest Labs' Freespace technology to allow users to change channels using gestures. As of early 2012, British LG SmartTV users using TalkTalk as in ISP had had significant service issues with the BBC iPlayer. LG have made no official comment, while the BBC have said that they are aware of the issue and are investigating.

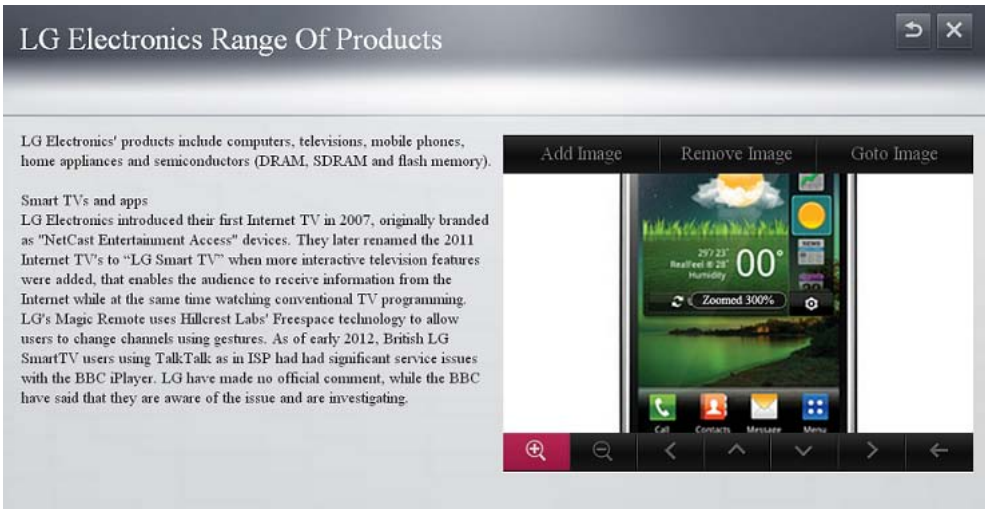
4) After pressing slide show button


LG Electronics Range Of Products

LG Electronics' products include computers, televisions, mobile phones, home appliances and semiconductors (DRAM, SDRAM and flash memory).

Smart TVs and apps
LG Electronics introduced their first Internet TV in 2007, originally branded as "NetCast Entertainment Access" devices. They later renamed the 2011 Internet TV's to "LG Smart TV" when more interactive television features were added, that enables the audience to receive information from the Internet while at the same time watching conventional TV programming. LG's Magic Remote uses Hillcrest Labs' Freespace technology to allow users to change channels using gestures. As of early 2012, British LG SmartTV users using TalkTalk as in ISP had had significant service issues with the BBC iPlayer. LG have made no official comment, while the BBC have said that they are aware of the issue and are investigating.

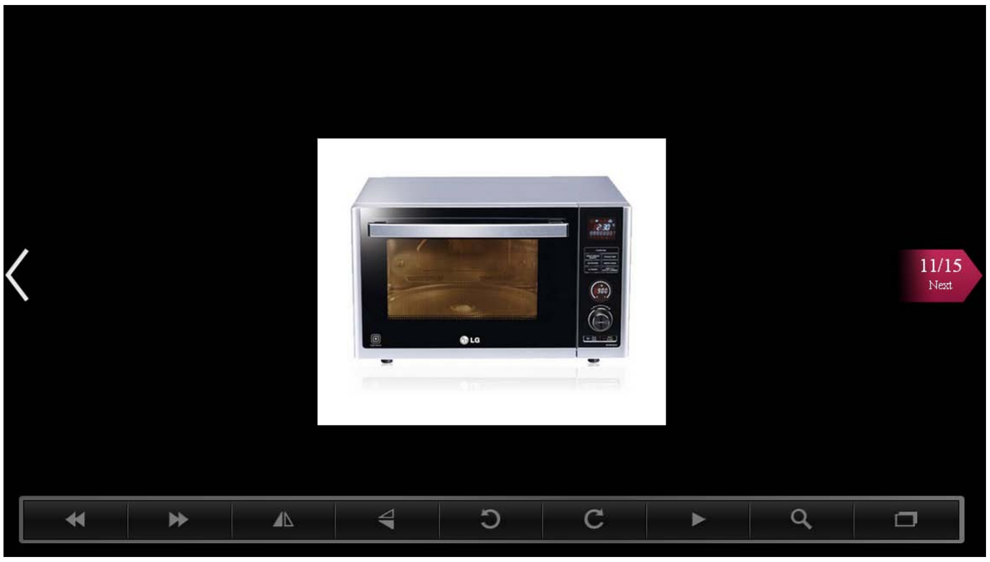
5) After pressing zoom then zoom in button


LG Electronics Range Of Products

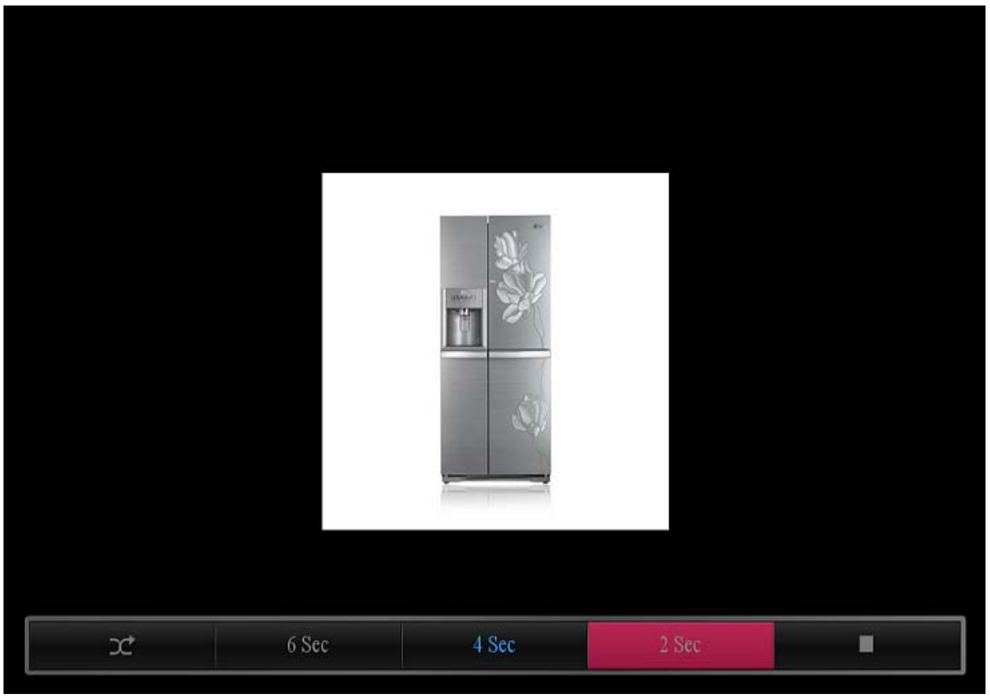
LG Electronics' products include computers, televisions, mobile phones, home appliances and semiconductors (DRAM, SDRAM and flash memory).

Smart TVs and apps
LG Electronics introduced their first Internet TV in 2007, originally branded as "NetCast Entertainment Access" devices. They later renamed the 2011 Internet TV's to "LG Smart TV" when more interactive television features were added, that enables the audience to receive information from the Internet while at the same time watching conventional TV programming. LG's Magic Remote uses Hillcrest Labs' Freespace technology to allow users to change channels using gestures. As of early 2012, British LG SmartTV users using TalkTalk as in ISP had had significant service issues with the BBC iPlayer. LG have made no official comment, while the BBC have said that they are aware of the issue and are investigating.

6) After pressing full Screen button



7) After pressing slide show button in Full Screen mode



4 Testig the Application

This chapter briefly describes how to test a developed application.

To debug the application on emulator, refer to **Testing > Testing App on Emulator** section in Developer Library in LG Developer (**http://developer.lge.com**) website.

To run and debug the application on real TV, refer to **Testing > Deploying and Testing App on Real TV** section in Developer Library in LG Developer (**http://developer.lge.com**) website.