

[Tutorial] LG Web_Integrating Video Advertisement into Web Application

Version 1.0 – October 2012

LGDEV-081

Home Entertainment Company
LG Electronics, Inc.

Copyright

Copyright © 2012 LG Electronics, Inc. All Rights Reserved.

Though every care has been taken to ensure the accuracy of this document, LG Electronics, Inc. cannot accept responsibility for any errors or omissions or for any loss occurred to any person, whether legal or natural, from acting, or refraining from action, as a result of the information contained herein. Information in this document is subject to change at any time without obligation to notify any person of such changes.

LG Electronics, Inc. may have patents or patent pending applications, trademarks copyrights or other intellectual property rights covering subject matter in this document. The provision of this document does not give the recipient or reader any license to these patents, trademarks copyrights or other intellectual property rights.

No part of this document may be communicated, distributed, reproduced or transmitted in any form or by any means, electronic or mechanical or otherwise, for any purpose, without the prior written permission of LG Electronics, Inc.

This document is subject to revision without further notice.

All brand names and product names mentioned in this document are trademarks or registered trademarks of their respective owners.

About This Document

Revision History

Document Version	Date	Comment
1.0	October 11, 2012	Initial Version

Purpose

This document describes how to integrate the video advertisements into web applications using LG Web Open API.

Reference Documents

Refer to the following documents:

- LG Web Application Development Guide
- LG Advertisement API Reference 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
AD	Advertisement
API	Application Programming Interface

Contents

1	Introduction.....	5
1.1	Overview	6
1.2	Needed APIs	7
2	Creating Application.....	8
2.1	Initializing the Page	9
2.2	Initializing the HTML5 Video.....	10
2.3	Handling Button Click Events	11
2.4	Handling VideoAd Status Events.....	13
2.5	Handling Playback Buttons.....	17
2.6	Inputting Keys	19
2.7	Setting HTML5 Video Object	20
2.8	Source Code of videoAd.html.....	21

Tables

[Table 1] Description of the Needed APIs.....	7
---	---

Figures

[Figure 1] Web Application using Video advertisement API.....	6
[Figure 2] Displaying VideoAd Status Event	16



1 Introduction

This chapter provides an overview of this application and needed APIs.

1.1 Overview

1.2 Needed APIs

1.1 Overview

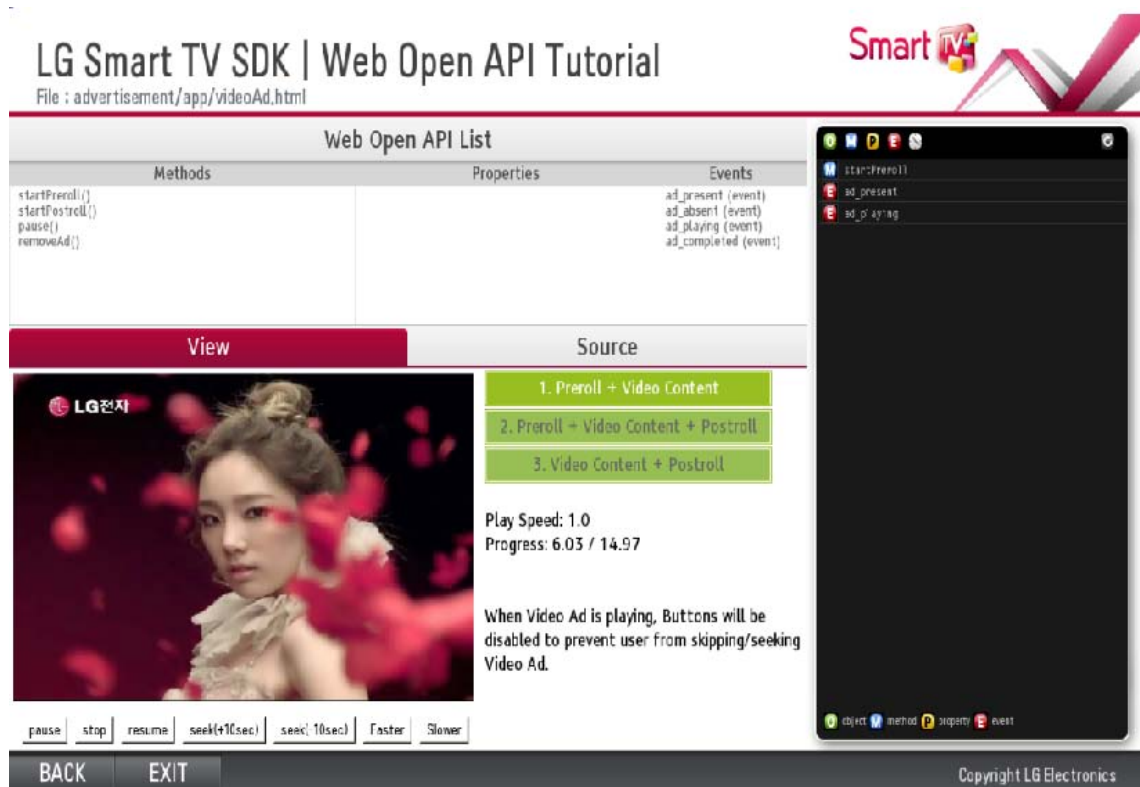
This application is designed to show which method, properties, and events of LG Web Open API are used to use video advertisement function on web applications in LG Smart TV. Supported ad types are video and banner.

Developers can integrate advertisements into their web applications using the Ad API that resides in the Smart Platform Library Server.

The VideoAd class is used to play a video advertisement. Currently, two types of video ad are available:

- Prefroll (before the main video)
- Postroll (after the main video)

Developers can choose the both (Before and after main video).



[Figure 1] Web Application using Video advertisement API

1.2 Needed APIs

This application uses following Web Open API:

[Table 1] Description of the Needed APIs

API Class	Name	Description
Method	startPreroll	Plays a video ad before the main video starts..
	startPostroll	Plays a video ad after the main video is completed.
	pause	Pauses the video ad currently being played..
	removeAd	To remove all the resources allocated for video ads.
Event	ad_present	Fired when there is a video ad to play.
	ad_absent	Fired when there is no video ad to play.
	ad_completed	Fired when the video ad is completed or the event is fired immediately after ad_absent event.
	ad_playing	Fired when the video ad is present and currently playing

For more information on these functions, refer to “LG Advertisement API Reference Guide”.

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 to use the video advertisement on web application using Web open API.

- 2.1 Initializing the Page
- 2.2 Initializing the HTML5 Video
- 2.3 Handling Button Click Events
- 2.4 Handling VideoAd Status Events
- 2.5 Handling Playback Buttons
- 2.6 Inputting Keys
- 2.7 Setting HTML5 Video Object
- 2.8 Source Code of videoAd.html

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.
- 06: Initialize the page.
- 07: Get the source code of the page using the XMLHttpRequest object.
- 08: Set the page id..
- 09: Initialize the Log function.
- 11: Call function to initialize voice.
- 14-20: Add event handlers which will be executed when the corresponding button is pressed.

Sample Code

```
01 : function initPage()
02 : {
03 :   //save as last visited page
04 :   setLastVisitPage();
05 :   //common initialize function
06 :   commonInitialize();
07 :   requestSourceCode();
08 :   setPageID("VideoAd");
09 :   jsLog.initLG();
10 :
11 :   initVideo();
12 :
13 :   //add onclick event handler
14 :   addEventHandler(document.getElementById("btn_back"), "click",
onClickHandler);
15 :   addEventHandler(document.getElementById("btn_exit"), "click",
onClickHandler);
16 :
17 :   addEventHandler(document.getElementById("preVideo"), "click",
handlePreroll);
18 :   addEventHandler(document.getElementById("preVideoPost"),
"click", handlePrepostroll);
19 :   addEventHandler(document.getElementById("videoPost"), "click",
handlePostroll);
20 : }
```

2.2 Initializing the HTML5 Video

- 03: Declare video element of HTML5.
- 04: Interval is set by using setInterval method of window class.

Sample Code

```
01 : function initVideo()  
02 : {  
03 :   video = document.getElementById("html5_video");  
04 :   setInterval(updatePlayProgress, updateInterval);  
05 : }
```

2.3 Handling Button Click Events

The following functions are event handlers which were added in **initPage** function.

handlePreroll

- 03: If adPlaying is true then return so that users should not able to skip or seek video ads.
- 04: Set adType variable to "preroll".
- 07: If scriptLoaded is true and adPlaying is false, declare videoAd and play the video advertisement. Where html5_video_div is the div tag id of the div that contains html5 video tag. html5_video is the video tag id events. prerollAdStatusListener is a call back function that listens to advertisement events. Content title is the title of the main video for ad targeting. 30 is the duration of pre/post roll in seconds.
- 08: Plays a video ad before the main video starts.
- 09: lgmethod to print log in console.
- 10: If Ad is completed then start playing main video content using playContent function.

handlePrepostroll

- 11: If adPlaying is true then return so that users should not able to skip or seek video ads.
- 13: Set adType variable to "prepostroll".
- 18-19: If scriptLoaded is true and adPlaying is false, declare videoAd and play the video advertisement.
- 20: lgmethod to print log in console.
- 31: else add EventListener for main video content to end and call the call back function handleContentEnd and then call the playContent function.

handle Postroll

- 27: If adPlaying is true then return so that users should not able to skip or seek video ads.
- 30: Set adType variable to "postroll".
- 31-32: Add EventListener to video and call the call back function handleContentEnd and then call the play content function

Sample Code

```

01 : function handlePreroll() {
02 :     if (adPlaying) {
03 :         return;
04 :     }
05 :     adType = "preroll";
06 :     if (scriptLoaded && (adPlaying == false)) {
07 :         videoAd = new VideoAd("html5_video_div",
"html5_video", prerollAdStatusListener, 'content title', 30);
08 :         videoAd.startPreroll();
09 :         jsLog.lgmethod('startPreroll');
10 :     } else playContent();
11 : }
12 : function handlePrepostroll() {
13 :     if (adPlaying) {
14 :         return;
15 :     }
16 :     adType = "prepostroll";
17 :     if (scriptLoaded && (adPlaying == false)) {
18 :         videoAd = new VideoAd("html5_video_div",
"html5_video", prepostrollAdStatusListener, 'hello kitty', 20);
19 :         videoAd.startPreroll();
20 :         jsLog.lgmethod('startPreroll');
21 :     } else {
22 :         video.addEventListener('ended', handleContentEnd, false);
23 :         playContent();
24 :     }
25 : }
26 : function handlePostroll() {
27 :     if (adPlaying) {
28 :         return;
29 :     }
30 :     adType = "postroll";
31 :     video.addEventListener('ended', handleContentEnd, false);

```

```
32 :           playContent();  
33 :       }
```

2.4 Handling VideoAd Status Events

Status listener for preroll Ad

This is a callback function that listens to advertisements events. This refers to ad before the main video.

- 04-06: If video ad is currently playing, disable videoPost and preVideoPost button so that users should not able to skip or seek video ads.
- 07: Set adPlaying as true.
- 10: If video ad is completed to play, enable videoPost and preVideoPost buttons.
- 11: Set adPlaying as false.
- 12: Remove all the resources allocated for video ads.
- 13: lgmethod to print log in console.
- 14: Calling play content function to play main video content.
- 15: Add EventListener for main video content to end and call the call back function handlePreContentEnd .
- 19: Once main video content ended remove EventListener for main video content to end.
- 20: Set videoPlaying as false.
- 21: Hide the progress status indicator.

Sample Code

```

01 : function prerollAdStatusListener(preEvent)
02 : {
03 :   jsLog.lgevent(preEvent);
04 :   if(preEvent == "ad_playing"){
05 :     document.getElementById("videoPost").className="executeBigButton
Off";
06 :     document.getElementById("preVideoPost").className="executeBigBut
tonOff";
07 :     adPlaying = true;
08 :   } else if (preEvent == "ad_completed"){
09 :     document.getElementById("videoPost").className="executeBigButton";
10 :     document.getElementById("preVideoPost").className="executeBigBut
ton";
11 :     adPlaying = false;
12 :     videoAd.removeAd();
13 :     jsLog.lgmethod('removeAd');
14 :     playContent();
15 :     video.addEventListener('ended', handlePreContentEnd, false);
16 :   }
17 : }
18 : function handlePreContentEnd() {
19 :     video.removeEventListener('ended', handlePreContentEnd, false);
20 :     videoPlaying = false;
21 :     statusIndicators.style.visibility= "hidden";
01 : }

```

Status listener for prepostroll Ad

This is a call back function that listens to advertisements events. This refers to ad before and after the main video.

- 04: If video ad is currently playing,disable preVideo and videoPost button so that users should not able to skip or seek video ads.
- 07: Set adplaying as true.
- 08-10: Else if video ad is completed enable preVideo and videoPost buttons.
- 11: Set adPlaying as false.
- 12: Remove all the resources allocated for video ads.
- 14: Add EventListener for main video content to end and call the call back function handleContentEnd.
- 15: Calling the playContent function.
- 19: Once main video content ended remove EventListener for main video content to end.
- 20-22: If scriptLoaded is true and adPlaying is false, declare videoAd and play the video advertisement

by using startPostroll function. where html5_video_div is the div tag id of the div that contains html5 video tag. html5_video is the video tag id events. postrollAdStatusListener is a call back function that listens to advertisement events. titanic is the title of the main video for ad targeting. 900 is the duration of post roll in seconds.

Sample Code

```
01 : function prepostrollAdStatusListener(preEvent)
02 : {
03 :   jsLog.lgevent(preEvent);
04 :   if(preEvent == "ad_playing"){
05 :     document.getElementById("preVideo").className="executeBigButtonOff";
06 :     document.getElementById("videoPost").className="executeBigButtonOff";
07 :     adPlaying = true;
08 :   }else if (preEvent == "ad_completed"){
09 :     document.getElementById("preVideo").className="executeBigButton";
10 :     document.getElementById("videoPost").className="executeBigButton";
11 :     adPlaying = false;
12 :     videoAd.removeAd();
13 :     jsLog.lgmethod('removeAd');
14 :     video.addEventListener('ended', handleContentEnd, false);
15 :     playContent();
16 :   }
17 : }
18 : function handleContentEnd() {
19 :   video.removeEventListener('ended', handleContentEnd, false);
20 :   if (scriptLoaded && (adPlaying == false)) {
21 :     videoAd = new VideoAd("html5_video_div",
22 :       "html5_video", postrollAdStatusListener, 'titanic', 900);
23 :     videoAd.startPostroll();
24 :     jsLog.lgmethod('startPostroll');
25 :   }
```

Status listener for postroll Ad

This is a callback function that listens to advertisement events. This refers to the ad after the main video.

- 04-11: If video ad is currently playing, in case of prepostroll disable preVideo and videoPost button and in case of post roll disable preVideo and prevideoPost button.
- 13: Set adplaying as true.
- 14-22: Else if video ad is completed playing, , in case of prepostroll enable preVideo and videoPost button and in case of post roll enable preVideo and prevideoPost.
- 24: Set adPlaying as false.
- 25: Remove all the resources allocated for video ads.
- 26: Making the source empty.
- 27: Updates the video element.
- 28: Set adPlaying as false..
- 29: Hide the progress status indicator.

Sample Code

```
01 : function postrollAdStatusListener(postEvent)
02 :   jsLog.lgevent(postEvent);
03 :   if(postEvent == "ad_playing"){
04 :     jsLog.lgmethod('postEvent ="ad_playing"');
05 :     if(adType=="prepostroll"){
06 :       document.getElementById("preVideo").className="executeBigButtonOff"
```

```

;
07 : document.getElementById("videoPost").className="executeBigButtonO
    ff";
08 :     }
09 :     if(adType=="postroll"){
10 :     document.getElementById("preVideo").className="executeBigButtonO
    ff";
11 :     document.getElementById("preVideoPost").className="executeBigBut
    tonOff";
12 :     }
13 :     adPlaying = true;
14 :     } else if (postEvent == "ad_completed") {
15 :         jsLog.lgmethod('postEvent ="ad_completed"');
16 :         if(adType=="prepostroll"){
17 :         document.getElementById("preVideo").className="executeBigButton"
            ;
18 :         document.getElementById("videoPost").className="executeBigButton";

19 :     }
20 :     if(adType=="postroll"){
21 :     document.getElementById("preVideo").className="executeBigButton"
            ;
22 :     document.getElementById("preVideoPost").className="executeBigBut
    ton";
23 :         }
24 :         adPlaying = false;
25 :         videoAd.removeAd();
26 :         jsLog.lgmethod('removeAd');
27 :         video.src = "";
28 :         video.load();
29 :         videoPlaying=false;
30 :         statusIndicators.style.visibility= "hidden";
31 :         }
32 :     }

```

To play the content

Content can be either audio or video.

- 03: mediaURL is set as the value for source.
- 04: Updates the video element.
- 05: Video is played.
- 06: Set videoPlaying to true.

Sample Code


```

01 : function playContent()
02 : {
03 :     video.src = mediaURL;
04 :     video.load();
05 :     video.play();
06 :     videoPlaying=true;
07 : }

```


LG Smart TV SDK | Web Open API Tutorial

File : advertisement/app/videoAd.html



Web Open API List

Methods	Properties	Events
startPreroll() startPostroll() pause() removeAd()		ad_present (event) ad_absent (event) ad_playing (event) ad_completed (event)



1. Preroll + Video Content

2. Preroll + Video Content + Postroll

3. Video Content + Postroll

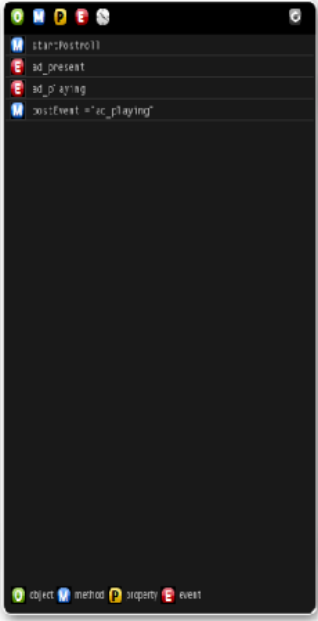
Play Speed: 1.0
Progress: 3.39 / 14.97

When Video Ad is playing, Buttons will be disabled to prevent user from skipping/seeking Video Ad.

pause stop resume seek(+10sec) seek(-10sec) Faster Slower

BACK EXIT

Copyright LG Electronics



[Figure 2] Displaying VideoAd Status Event

2.5 Handling Playback Buttons

These are the buttons to resume, pause, stop, seek and to change the speed of the main video content. Only Pause and resume will work for video advertisement.

- 01: handleResume function.
- 02: Video is played.
- 03: Set videoPlaying to true.

- 05: handlePause function
- 06: Video is paused.
- 07: Set videoPlaying to false.
- 08: If ad is playing then that videoAd is paused.

- 13: handleStop function.
- 14: The current playing video is paused.
- 15: Set videoPlaying to false.
- 16: Updates the video element.
- 17: Hide the progress status bar.

- 20: handleSeek function
- 22: If the sum of video's current time and seconds added to it is lesser than the video duration then display the time after the added seconds to the current time of the video else display the video duration.

- 25: handleChangeSpeed function.
- 26: Increment playback rate by respective value.

- 28: updatePlayProgress function.
- 29: If adPlaying or videoPlaying is true then make progress status bar visible.
- 31-32: On progress of the video.the play speed and the current time will update itself on progress of the video.
- 33-34: Display new value to status bar.
- 35: Hide the status bar if both are not playing.

Sample Code

```

01 :   function handleResume(){
02 :   video.play();
03 :   videoPlaying=true;
04 :   }
05 :   function handlePause() {
06 :       video.pause();
07 :       videoPlaying=false;
08 :       if(adPlaying){
09 :           videoAd.pause();
10 :       jsLog.lgmethod('pause');
11 :       }
12 :   }
13 :   function handleStop(){
14 :   video.pause();
15 :   videoPlaying=false;
16 :   video.src="";
17 :   video.load();
18 :   statusIndicators.style.visibility= "hidden";
19 :   }
20 :   function handleSeek(secFromNow){
21 :   if(video.currentTime+secFromNow<video.duration){
22 :   video.currentTime += secFromNow;
23 :   }else video.currentTime=video.duration;
24 :   }
25 :   function handleChangeSpeed(rateIncBy){
26 :   video.playbackRate += rateIncBy;
27 :   }
28 :   function updatePlayProgress (){

```

```
29 :   if(adPlaying||videoPlaying){
30 :     statusIndicators.style.visibility= "visible";
31 :     var playspeedText = "Play Speed: " +
      video.playbackRate.toFixed(1);
32 :     var ProgressText = "Progress: " + video.currentTime.toFixed(2) +
      " / " + video.duration.toFixed(2);
33 :     setInnerTextById("playspeed", playspeedText);
34 :     setInnerTextById("currentTime", ProgressText);
35 :   }else statusIndicators.style.visibility= "hidden";}
```

2.6 Inputting Keys

Use **onUserInput** function is called by **onClickHandler** function; it receives a key value as the **userInput** parameter from **onClickHandler** and creates the corresponding function for each key value to operate the key.

05: When the back key is pressed, it will open previous page.

Sample Code

```
01 : function onUserInput(userInput)
02 : {
03 :   switch(userInput)
04 :   {
05 :     case VK_BACK :
06 :       window.location.replace("../menu_advertisement.html"); break;
07 :     break;
08 :   }
```

2.7 Setting HTML5 Video Object

The following code shows how to set video advertisement object.

- 01: Set id value as 'html5_video'.
- 02-03: Setting default properties i.e. poster, width and height.
- 04: Setting Message for browsers which is not supporting video tag

Sample Code

```
01 : <video id="html5_video"  
02 :   poster=""  
03 :   width=520 height=300>  
04 :   <i> Browser doesn't support video tag</i>  
05 : </video>
```

2.8 Source Code of videoAd.html

Source code of videoAd.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>Advertisement API Test</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 type="text/javascript" onload='scriptLoaded=true' onerror='scriptLoaded=false'
src="http://smartservice.lgappstv.com/library/apps/ad/lib/videoAd.js"></script>
<script type="text/javascript">
    var mediaURL = "http://smartservice.lgappstv.com/samples/videoAd/media/test.mp4";
    var video = null;
    var videoAd = null;
    var updateInterval = 100;
    var adPlaying = false;//a boolean variable to determine whether video being played is Ad
    var videoPlaying = false;//a boolean variable to determine whether video is playing
    var adType = "";
    //initialize page
    function initPage() {
        //save as last visited page
        setLastVisitPage();

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

        initVideo();

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

        addEventHandler(document.getElementById("preVideo"), "click", handlePreroll);
```

```
        addEventHandler(document.getElementById("preVideoPost"), "click", handlePrepostroll);
        addEventHandler(document.getElementById("videoPost"), "click", handlePostroll);
    }

    function onUserInput(userInput) {
        switch (userInput) {
            case VK_BACK:
                window.location.replace("../menu_advertisement.html");
                break;
        }
    }

    function initVideo() {
        video = document.getElementById("html5_video");
        setInterval(updatePlayProgress, updateInterval);
    }

    function handlePreroll() {
        if (adPlaying) {
            return;
        }
        adType = "preroll";
        if (scriptLoaded && (adPlaying == false)) {
            videoAd = new VideoAd("html5_video_div", "html5_video", prerollAdStatusListener, 'content title', 30);
            videoAd.startPreroll();
            jsLog.lgmethod('startPreroll');
        } else playContent();
    }

    function prerollAdStatusListener(preEvent) {
        jsLog.lgevent(preEvent);
        if (preEvent == "ad_playing") {
            document.getElementById("videoPost").className = "executeBigButtonOff";
            document.getElementById("preVideoPost").className = "executeBigButtonOff";
            adPlaying = true;
        } else if (preEvent == "ad_completed") {
            document.getElementById("videoPost").className = "executeBigButton";
            document.getElementById("preVideoPost").className = "executeBigButton";
            adPlaying = false;
            videoAd.removeAd();
            jsLog.lgmethod('removeAd');
            playContent();
            video.addEventListener('ended', handlePreContentEnd, false);
        }
    }
```

```

    }
}

function handlePreContentEnd() {
    video.removeEventListener('ended', handlePreContentEnd, false);
    videoPlaying = false;
    statusIndicators.style.visibility= "hidden";
}

function handlePrepostroll() {
    if (adPlaying) {
        return;
    }
    adType = "prepostroll";
    if (scriptLoaded && (adPlaying == false)) {
        videoAd = new VideoAd("html5_video_div", "html5_video", prepostrollAdStatusListener, 'hello kitty', 20);
        videoAd.startPreroll();
        jsLog.lgmethod('startPreroll');
    } else {
        video.addEventListener('ended', handleContentEnd, false);
        playContent();
    }
}

function prepostrollAdStatusListener(preEvent) {
    jsLog.lgevent(preEvent);
    if (preEvent == "ad_playing") {
        document.getElementById("preVideo").className = "executeBigButtonOff";
        document.getElementById("videoPost").className = "executeBigButtonOff";
        adPlaying = true;
    } else if (preEvent == "ad_completed") {
        document.getElementById("preVideo").className = "executeBigButton";
        document.getElementById("videoPost").className = "executeBigButton";
        adPlaying = false;
        videoAd.removeAd();
        jsLog.lgmethod('removeAd');
        video.addEventListener('ended', handleContentEnd, false);
        playContent();
    }
}

function handleContentEnd() {
    video.removeEventListener('ended', handleContentEnd, false);
    if (scriptLoaded && (adPlaying == false)) {

```

```
        videoAd = new VideoAd("html5_video_div", "html5_video", postrollAdStatusListener, 'titanic', 900);
        videoAd.startPostroll();
        jsLog.lgmethod('startPostroll');
    }
}

function postrollAdStatusListener(postEvent) {
    jsLog.lgevent(postEvent);
    if(postEvent == "ad_playing"){
        jsLog.lgmethod('postEvent = "ad_playing"');
        if(adType=="prepostroll"){
            document.getElementById("preVideo").className="executeBigButtonOff";
            document.getElementById("videoPost").className="executeBigButtonOff";
        }

        if(adType=="postroll"){
            document.getElementById("preVideo").className="executeBigButtonOff";
            document.getElementById("preVideoPost").className="executeBigButtonOff";
        }

        adPlaying = true;
    } else if (postEvent == "ad_completed") {
        jsLog.lgmethod('postEvent = "ad_completed"');
        if(adType=="prepostroll"){
            document.getElementById("preVideo").className="executeBigButton";
            document.getElementById("videoPost").className="executeBigButton";
        }

        if(adType=="postroll"){
            document.getElementById("preVideo").className="executeBigButton";
            document.getElementById("preVideoPost").className="executeBigButton";
        }

        adPlaying = false;
        videoAd.removeAd();
        jsLog.lgmethod('removeAd');
        video.src = "";
        video.load();
        videoPlaying=false;
        statusIndicators.style.visibility= "hidden";
    }
}

function handlePostroll() {
    if (adPlaying) {
        return;
    }
    adType = "postroll";
    video.addEventListener('ended', handleContentEnd, false);
}
```



```
        playContent();
    }

    function playContent() {
        video.src = mediaURL;
        video.load();
        video.play();
        videoPlaying=true;
    }

    function handleResume() {
        video.play();
        videoPlaying=true;
    }

    function handlePause() {
        video.pause();
        videoPlaying=false;
        if (adPlaying) {
            videoAd.pause();
            jsLog.lgmethod('pause');
        }
    }

    function handleStop() {
        video.pause();
        videoPlaying=false;
        video.src = "";
        video.load();
        statusIndicators.style.visibility= "hidden";
    }

    function handleSeek(secFromNow) {
        if (video.currentTime + secFromNow < video.duration) {
            video.currentTime += secFromNow;
        } else video.currentTime = video.duration;
    }

    function handleChangeSpeed(rateIncBy) {
        video.playbackRate += rateIncBy;
    }

    function updatePlayProgress() {
```

```
        if(adPlaying||videoPlaying){
            statusIndicators.style.visibility= "visible";
            var playspeedText = "Play Speed: " + video.playbackRate.toFixed(1);
            var ProgressText = "Progress: " + video.currentTime.toFixed(2) + " / " + video.duration.toFixed(2);
            setInnerTextById("playspeed", playspeedText);
            setInnerTextById("currentTime", ProgressText);
        }else statusIndicators.style.visibility= "hidden";
    }
</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 : advertisement/app/videoAd.html</div>
    </div>
    <div class='SuiteTitleLine'></div>
    <!-- test contents -->
    <div id='content_body' class='ContentArea'>
        <div class='ApiListTitleArea'>Web Open API List</div>
        <div class='ApiListArea'>
            <div class='MethodTitleArea'>Methods
                <div class='MethodListArea'>startPreroll()
                    <br>startPostroll()
                    <br>pause()
                    <br>removeAd()
                </div>
            </div>
            <div class='PropertyTitleArea'>Properties
                <div class='PropertyListArea'></div>
            </div>
            <div class='EventTitleArea'>Events
                <div class='EventListArea'>ad_present (event)
                    <br>ad_absent (event)
                    <br>ad_playing (event)
                    <br>ad_completed (event)
                </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>
</body>
```

```

<div id='view' class='ViewArea'>
  <table border="0">
    <tr>
      <td rowspan="4" width="520" height="300">
        <div id="html5_video_div" style="top: 5px; left: 5px; width: 520px; height: 300px; float:
left; position: absolute; background-color:black;">
          <video id="html5_video" poster="" width=520 height=300>      <i> Browser doesn't support
video tag</i> </video>
        </div>
      </td>
      <td>
        <div id="preVideo" class="executeBigButton">1. Preroll + Video Content</div>
      </td>
    </tr>
    <tr>
      <td>
        <div id="preVideoPost" class="executeBigButton">2. Preroll + Video Content + Postroll</div>
      </td>
    </tr>
    <tr>
      <td>
        <div id="videoPost" class="executeBigButton">3. Video Content + Postroll</div>
      </td>
    </tr>
    <tr>
      <td height="60px">
        <div id="statusIndicators" style="visibility:hidden">
          <div id="playspeed"></div>
          <div id="currentTime"></div>
        </div>
        <div>
          <br>
          <br>When Video Ad is playing, Buttons will be disabled to prevent user from
skipping/seeking Video Ad.
        </div>
      </td>
    </tr>
    <tr>
      <td width="520" height="50">
        <div id="controlButtonArea">
          <button id="pause" onclick="handlePause()">pause</button>
          <button id="stop" onclick="handleStop()">stop</button>
          <button id="resume" onclick="handleResume()">resume</button>
          <button id="seekPlus5" onclick="handleSeek(10)">seek(+10sec)</button>
          <button id="seekMinus5" onclick="handleSeek(-10)">seek(-10sec)</button>
        </div>
      </td>
    </tr>
  </table>
</div>

```

```
                <button id="incPlaySpeed" onclick="handleChangeSpeed(0.5)">Faster</button>
                <button id="decPlaySpeed" onclick="handleChangeSpeed(-0.5)">Slower</button>
            </div>
        </td>
    </tr>
</table>
</div>
<div style="visibility: hidden" id='codeview'>
    <textarea class="SourceCodeArea" value="" id='sourcecode'></textarea>
</div>
</div>
<!-- description -->
<div class='SuiteDescription' id='back_exit_test_description'></div>
<!-- button and copyright -->
<div class='SuiteButtonArea'>
    <!-- back key description -->
    <div id='btn_back' class='buttonDescription'>BACK</div>
    <!-- exit button -->
    <div id='btn_exit' class='buttonDescription'>EXIT</div>
    <!-- copyright -->
    <div class='copyright'>Copyright LG Electronics</div>
</div>
</body>
</html>
```