Client (HHP) to TV Application Communication

Published 2014-10-28 | (Compatible with SDK 3.5,4.5,5.0,5.1 and 2012,2013,2014 models)

Contents

REST Interface

The communication path from the client device (HHP) to the TV application is provided via the REST interface. The following calls are designed to be a generic set of communication APIs that allow developers to define the content of the messages, while providing a structured protocol easily understood by the server (TV).

Each of the following requests includes a URI that contains an {appID}, which is required to communicate with the corresponding TV application.

Request Headers

Header	Description	Required
SLDeviceID	This header contains the DeviceID used to identify the client device and the corresponding TV application.	**YES** (for ALL requests except Get Application Info request)
Content-Type	This header refers to the content type of the request. The value for this header can be either application/json or text/xml.	**YES** (for POST requests only)
ProductID	This header is only required for Device Connect requests. The value MUST begin with “SMART” and be exactly 8 characters long. Example: SMARTTVI	**YES** (for Connect requests only)
VendorID	This header is only required for Device Connect requests. The value must be exactly 8 characters long.	**YES** (for Connect requests only)
DeviceName	This header is only required for Device Connect requests. This header contains the name of the client device. The value MUST be less than 64 characters long.	**YES** (for Connect requests only)
User-Agent	This header is used by the TV to identify the type of the client device and its capabilities.	**YES** (for ALL requests)
Custom (User-Defined)	The developer can add any other headers needed for proper performance of their application. Any custom headers added to a request are included as the second parameter in the message context (JSON object).	**NO**

Application Methods

Get Application Info

This request retrieves the following information related to the application ID included in the request. This is a

system message and the message format is predefined in JSON format.

HTTP method and URI GET /ws/app/{appID}/info	
Syntax	**Request type:** GET **URI:** /ws/app/{appID}/info
Return value	200 OK 404 Not found Body: Widget information presented as JSON. For example: {"widgetInfo": {"ID": "Canvas_NewOCI", "Version": "0.930", "Status": "RUNNING", "widgetname": "sampleWidget"}}
Parameter	appID (Application ID)
Example	**Sample request:** GET /ws/app/sampleWidget/info HTTP/1.1 User-Agent: curl/7.20.1 (i686-pc-cygwin) libcurl/7.20.1 OpenSSL/0.9.8r zlib/1.2.5 libidn/1.18 libssh2/1.2.5 Host: 192.168.1.108 Accept: */* **Sample response:** HTTP/1.1 200 OK Content-Type: text/html Transfer-Encoding: chunked Date: Thu, 14 Jul 2011 22:16:25 GMT Server: lighttpd/1.4.28 {"widgetInfo": {"ID": "Canvas_NewOCI", "Version": "0.930", "Status": "RUNNING", "widgetname": "sampleWidget"}}

Connect to Application

This request initiates the connection between the client application and the TV application. Client must provide the following headers with the request:

SLDeviceID: Client's Device ID. The ID is translated by TV M/W and there will be 1-to-1 mapping to device ID passed to TV application.

VenderID: Vendor ID. It should be provided by Samsung and should be 8 characters exactly.

ProductID: Product ID. It should be provided by Samsung and should be 8 characters exactly.

HTTP method and URI POST /ws/app/{appID}/connect	
Syntax	**Request Type:** POST **URI:** /ws/app/{appID}/connect
Return value	200 OK (connection request is accepted) 403 Forbidden 404 Not found
Parameter	appID (Application ID)

HTTP method and URI POST /ws/app/{appID}/connect	
Example	**Sample request:** POST /ws/app/sampleWidget/connect HTTP/1.1 Accept: */* Accept-Language: en-us SLDeviceID: 12345 VendorID: VenderMe DeviceName: IE-Client ProductID: SMARTDev Accept-Encoding: gzip, deflate User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0) Host: 127.0.0.1:8080 Content-Length: 0 Connection: Keep-Alive **Sample response:** HTTP/1.1 200 OK Content-Length: 0 Date: Thu, 14 Jul 2011 3:32 PM Server: lighttpd/1.4.28

Disconnect from Application

This request disconnects the mobile application from the TV application.

HTTP method and URI POST /ws/app/{appID}/disconnect	
Syntax	**Request type:** POST **URI:** /ws/app/{appID}/disconnect
Return value	200 OK (disconnect request is accepted)
Parameter	appID (Application ID)
Example	**Sample request:** POST /ws/app/sampleWidget/disconnect HTTP/1.1 Accept-Encoding: gzip, deflate Accept-Language: en-us SLDeviceID: 12345 Accept: */* User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0) Host: 192.168.1.108 Content-Length: 0 Connection: Keep-Alive Pragma: no-cache **Sample response:** HTTP/1.1 200 OK Content-Length: 0 Date: Thu, 01 Jan 1970 01:32:23 GMT Server: lighttpd/1.4.28

Message Queue Methods

Each of the following requests is designed to communicate directly with the message queueing system on the TV. These REST APIs are limited to either pushing a message onto the queue or popping (removing) a message from the queue.

Message from Mobile App to TV App

This request pushes a TV application message onto the message queue in the TV. Any message sent to the following URI only goes to the TV application.

> Note
>
> The message body included in this request is determined by the developer of the

application.

HTTP method and URI POST /ws/app/{appID}/queue Push message (to TV App): To send application data to TV App	
Syntax	**Request type:** POST **URI:** /ws/app/{appID}/queue
Return value	200 OK (message sent to application) 403 Forbidden (client not allowed to send message to the requested application) 404 Not found (requested application does not exist)
Parameter	appID (Application ID)
Example	**Sample request:** POST /ws/app/sampleWidget/queue HTTP/1.1 Accept: application/json Accept-Language: en-us SLDeviceID: 12345 Content-Type: application/json User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0) Host: 192.168.1.108 Content-Length: 56 {"type":"touchMove","coordinates":{"x":"343","y":"252"}} **Sample response:** HTTP/1.1 200 OK Content-Length: 0 Date: Thu, 14 Jul 2011 3:32 PM Server: lighttpd/1.4.28

Pop Message

If the message queue is not empty, this request retrieves a message from the device's message queue on the TV. If the message queue is empty, this request is be held until the message arrives or the request times out (long polling). Status code 408 is returned upon long polling time out. The client must initiate another pop message request immediately when it receives 408.

HTTP method and URI GET /ws/app/{appID}/queue/devices/{deviceID} Push message (from TV App): To get application data from TV App	
Syntax	**Request type:** GET **URI:** /ws/app/{appID}/queue/devices/{deviceID}
Return value	200 OK (message sent to application) 403 Forbidden (client not allowed to send message to the requested application) 404 Not found (requested application does not exist) 408 Time out (long polling time out)
Parameter	appID (Application ID) deviceID (Device ID)
Example	**Sample request:** GET /ws/app/Canvas_NewOCI/queue/devices/12345 HTTP/1.1 User-Agent: curl/7.20.1 (i686-pc-cygwin) libcurl/7.20.1 OpenSSL/0.9.8r zlib/1.2.5 libidn/1.18 libssh2/1.2.5 Host: 127.0.0.1:8080 Accept: application/json SLDeviceID:12345 **Sample response:** HTTP/1.1 200 OK Content-Type: application/json sender: TV Transfer-Encoding: chunked Date: Fri, 15 Jul 2011 21:01:59 GMT Server: lighttpd/1.4.28 {"type":"echo","coordinates":{"x":"343","y":"252"}}

Device Group Methods

Join Group

This request is sent to inform the TV that the device wants to join a particular group.

HTTP method and URI POST /ws/app/{appID}/queue/groups/{groupID}/join	
Syntax	**Request type:** POST **URI:** /ws/app/{appID}/queue/groups/{groupID}/join
Return value	200 OK (joining request accepted) 403 Forbidden (client does not have the permission to execute the specified operation) 404 Not found (requested group does not exist)
Parameter	appID (Application ID) groupID (Group ID) {groupID} corresponds to the group ID the client device wishes to join.
Example	**Sample request:** POST /ws/app/sampleWidget/queue/groups/feigroup/join HTTP/1.1 SLDeviceID:12345 User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0) Content-Length: 0 **Sample response:** HTTP/1.1 200 OK Content-Length: 0 Date: Thu, 14 Jul 2011 3:32 PM Server: lighttpd/1.4.28

Leave Group

This request is sent to inform the TV that the device wants to leave a particular group.

HTTP method and URI POST /ws/app/{appID}/queue/groups/{groupID}/leave	
Syntax	**Request type:** POST **URI:** /ws/app/{appID}/queue/group/{groupID}/leave
Return value	200 OK (leaving request accepted) 403 Forbidden (client does not have the permission to execute the specified operation) 404 Not found (requested group does not exist)
Parameter	appID (Application ID) groupID (Group ID) {groupID} corresponds to the group ID the client device wishes to leave.
Example	**Sample request:** POST /ws/app/sampleWidget/queue/groups/feigroup/leave HTTP/1.1 SLDeviceID:12345 User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0) Content-Length: 0 **Sample response:** HTTP/1.1 200 OK Content-Length: 0 Date: Thu, 14 Jul 2011 3:32 PM Server: lighttpd/1.4.28

List Group Members

This request retrieves a list of all devices currently joined to a particular group.

HTTP method and URI GET /ws/app/{appID}/queue/groups/{groupID}	
Syntax	**Request type:** GET **URI:** /ws/app/{appID}/queue/group/{groupID}
Return value	200 OK (request is fulfilled and the result is in the body) 403 Forbidden (client does not have the permission to execute the specified operation) 404 Not found (requested group does not exist) BODY

HTTP method and URI GET /ws/app/{appID}/queue/groups/{groupID}	
Parameter	appID (Application ID) groupID (Group ID)
Example	**Sample request:** GET /ws/app/sampleWidget/queue/groups/Canvas_NewOCI HTTP/1.1 User-Agent: curl/7.20.1 (i686-pc-cygwin) libcurl/7.20.1 OpenSSL/0.9.8r zlib/1.2.5 libidn/1.18 libssh2/1.2.5 Host: 127.0.0.1:8080 Accept: application/json SLDeviceID:12345 **Sample response:** HTTP/1.1 200 OK Content-Type: application/json Transfer-Encoding: chunked Date: Fri, 15 Jul 2011 21:10:10 GMT Server: lighttpd/1.4.28 {"group":{"Devices":[{"SLDeviceID":"12345"},]}}

File Upload Methods

It is also possible to send files to the corresponding TV application. In these requests, the content type must be multipart form data. Multiple messages and files may be sent together. Each message has a separate callback on the TV application. Client devices can then access the uploaded files through the following URI: /ws/app/{appID}/file/{filename} .

The size limit for single uploaded file and total uploaded files is 3MB. All the uploaded files will be cleaned up when the TV application quits execution.

Upload File

This request sends a message containing the file data to the message queue on the TV.

Content type must be Multipart form data. Multiple messages and files can be sent together. Each message will be a separate callback on application. Files are saved and served through URI: /ws/app/{appID}/file/{filename} .

HTTP method and URI POST /ws/app/{appID}/queue	
Syntax	**Request type:** POST **URI:** /ws/app/{appID}/queue
Return value	200 OK
Parameter	appID (Application ID)
Example	**Sample request:** POST /ws/app/sampleWidget/queue HTTP/1.1 User-Agent: curl/7.20.1 (i686-pc-cygwin) libcurl/7.20.1 OpenSSL/0.9.8r zlib/1.2.5 libidn/1.18 libssh2/1.2.5 Host: 192.168.1.108 Accept: */* SLDeviceID: 12345 Content-Length: 13774 Expect: 100-continue Content-Type: multipart/form-data; Content-Disposition: form-data; name="message" {"type": "ShowPic", "filename": "UE-1.jpg"} Content-Disposition: form-data; name="upload"; filename="UE-1.jpg" Content-Type: image/jpeg ... **Sample response:** HTTP/1.1 200 OK Content-Length: 0 Date: Thu, 14 Jul 2011 3:32 PM Server: lighttpd/1.4.28

Delete File

This request will delete a file on the TV.

HTTP method and URI DELETE /ws/app/{appID}/file/{filename}	
Syntax	**Request type:** DELETE **URI:** /ws/app/{appID}/file/{filename}
Return value	200 OK 204 HTTP_NOCONTENT (there is no file) 404 HTTP_NOTFOUND (app id is not found)
Parameter	appID (Application ID) filename (File name to delete)
Example	**Sample request:** POST /ws/app/sampleWidget/file/test1.jpg HTTP/1.1 User-Agent: curl/7.20.1 (i686-pc-cygwin) libcurl/7.20.1 OpenSSL/0.9.8r zlib/1.2.5 libidn/1.18 libssh2/1.2.5 Host: 192.168.1.108 Accept: */* SLDeviceID: 12345 ... ... **Sample response:** HTTP/1.1 200 OK Content-Length: 0 Date: Thu, 14 Jul 2011 3:32 PM Server: lighttpd/1.4.28