Sony Bravia IP plugin

Document Version 1:
31 March 2020
By A-Lurker

License

Sony API

Further information on the Sony Bravia REST API can be found here:

https://pro-bravia.sony.net/develop/integrate/rest-api/spec/index.html

And the InfraRed Compatible Control (IRCC) information here:

https://pro-bravia.sony.net/develop/integrate/ircc-ip/overview/index.html

Test bed

Tested using openLuup and AltUI on a Sony Bravia "KDL 65W850C" TV. Model year: 2015.

There are any number of Bravia TV models, so the plugin may give different results from what are described here. Additionally some functionality may not. However this site suggest that TVs as far back as 2011 may work?

https://info.tvsideview.sony.net/en_ww/home_device.html#bravia

TV set up

The TV needs to be set up appropriately. In particular you need to make up and enter a **Pre-Shared Key** into the TV and the plugin. Note that if you update the TV's firmware at any stage, these settings may be altered and or reset:

- Set up a static IP for the TV
- Enable Wake on LAN on your TV: Settings → Network → Home Network Setup → Remote Start → On
- Enable pre-shared key on your TV: Settings → Network → Home Network Setup → IP Control → Authentication → Normal and Pre-Shared Key
- Set pre-shared key on your TV: Settings → Network → Home Network Setup → IP Control → Pre-Shared Key → Enter your own PSK
- Enable Remote device/Renderer: Settings → Network → Home Network Setup → Remote device/Renderer → Remote device/Renderer → On
- Enable Remote device/Renderer: Settings → Network → Home Network Setup → Remote device/Renderer → Remote access control → Auto access permission → on

Installation

Under AltUI:

While showing the 'Devices' page hit the '+Create' button. In the appropriate entry boxes, insert the following:

- A device name you decide upon eg Sony Bravia
- D_Sony_Bravia_IP1.xml
- I_Sony_Bravia_IP1.xml

Save the changes.

Restart the Luup Engine
Restart AltUI

When the device appears, go to the variables and insert and save the following:

IP: insert the TV IP address
PSK: insert the Pre-Shared Key (PSK) you set up in the TV.

Restart the Luup Engine
Restart AltUI

Not working: keep trying till functional! Also refer to the DebugEnabled flag further below.

Web page

You can access a status report for the TV by looking at the plugin's TV information web page: refer to the web page link on 'Control' Tab in the UI for the plugin. See also the end of this document for examples:

Alternatively enter this URL into your browser:

http://Veras_IP_address:3480/port_3480/data_request?id=lr_al_sony_bravia_info

At the top of the web page is a text entry box and submit button. It can be used instead of the incredibly annoying on screen virtual keyboard. So for example, if the TV is showing its web browser application and you want to enter a URL; you can use the web page entry box and submit button to replicate the actions of a on screen virtual keyboard.

General operation

- If the TV is powered completely off; a WOL packet maybe be required to get it started. The TV options need to be adjusted appropriately to allow WOL packets to be acted upon.

- This plugin code requires the TV to be in 'Standby' mode or at some other higher power level.

IR codes sent using IRCC

The plugin web page lists all the remote codes available. Some notes about the IR codes:

- Different TV models may have different IR code sets. ie not all IR codes are always available.

- Some IR codes are replicated, so 'Sleep" and PowerOff' may be the same actual IR code

- Other examples of duplicates:
  Audio   & MediaAudioTrack
  PicOff & PictureOff
  Analog2 & TvAnalog
  Input   & TvInput
  Num12   & Enter

- The actual sent IR codes are case sensitive. So these are different:
  'AAAAAQAAAAEAAAAUAw==' is Mute and 'AAAAAQAAAAEAAAAuAw==' is WakeUp

- It is somewhat unclear what some codes do eg:
  '*AD'
  'FeaturedAppVOD'
  'PAP'   (Picture-and-Picture)
  'ShopRemoteControlForcedDynamic'

Summary of On Off IR power codes

Summary of IR power codes:	
PowerOff	status becomes standby: same code as Sleep
Sleep	status becomes standby: same code as PowerOff
WakeUp	status becomes: active
TvPower	status active or standby. THIS is a toggle

Summary of potential content sources

There are potentially two types of media sources. The web page can be used to report these and in the case of the applications, start them.

1) Applications such as YouTube, Netfilx, etc invoked using SetActiveApp

2) Actual physical sources selected using SetPlayContent as listed below.

Typical physical TV sources selectable with SetPlayContent:	
See also: https://pro-bravia.sony.net/develop/integrate/rest-api/spec/resource-uri-list/index.html	
tv:dvbc	Digital Video Broadcasting - Cable
tv:dvbs	Digital Video Broadcasting - Satellite
tv:dvbt	Digital Video Broadcasting - Terrestrial
Typical external sources:	
extInput:hdmi	HDMI (High-Definition Multimedia Interface) Can also be: MHL (Mobile High-Definition Link) Also HDMI ARC (Audio Return Channel)
extInput:composite	Composite
extInput:component	Component
extInput:cec	CEC (Consumer Electronics Control)
extInput:scart	Syndicat des Constructeurs d'Appareils Radiorécepteurs et Téléviseurs
extInput:widi	WiFi eg “Screen Mirroring”
fav:tv	Favourite TV channels or Applications
usb:recStorage	USB input

Notes:
With HDMI ARC you can connect everything to the TV. The TV can then send all audio down the HDMI cable to your high quality ARC capable AVR set up. The TV audio however may be two-

channel (2.0) only via ARC, instead of say 5.1. Sony specs are unclear in this regard.

The plugin service id

```
urn:schemas-a-lurker-com:service:Sony_Bravia_IP:1
```

Did I mention that alphabetic case is important?

Attributes, variables & functions

The plugin has the following attributes, variables & functions:	
Variables:	
Connected	0 or 1
DebugEnabled	0 or 1
DisplayIsOn	0 or 1
IP	The TV's IP address
MAC	The TV's MAC address – populated automatically once the TV is 'connected'
ModelName	Eg: KDL-65W850C
Mute	0 or 1
PluginVersion	0.51
Volume	0 to 100
Functions:	
ExecuteMethod	Variables: Method, Json, Service For developers: requires all these variables to be supplied
SendRemoteCode	Variables: RemoteCode Name or value of IR code eg: 'WakeUp' or 'AAAAAQAAAAEAAAAuAw' and is **case sensitive**
SendWOL	Variables: None Send Wake On LAN. TV settings need to be correct to respond to WOL. Also MAC must be set
SetActiveApp	Variables: Uri Launch an application eg Netflix, YouTube, Browser, etc. Specify the associated URI
SetMute	Variables: Mute 0, 1 and T for toggle
SetPlayContent	Variables: Uri Enter a valid source name eg: 'extInput:hdmi?port=1' and is **case sensitive**
SetPower	Variables: Power 0 or 1 Switches between 'Standby' & 'Active'. Power status is the DisplayIsOn variable
SetTextForm	Variables: Text Any 'text' to enter into the infuriating onscreen virtual keyboard entry box
SetVolume	Variables: Volume Integer: 0 to 100
SetVolumeUpDown	Variables: Step Integer: +/- 2, 5 or 10
TerminateApps	Variables: None Terminate any running application(s) eg Netflix, YouTube, Browser, etc

Notes:

- The plugin maintains its own IP and MAC variables.
- The Volume and Mute can't be altered if the display is off. If you are unsure if the display is off or on then check the DisplayIsOn variable first. The display is considered to be off in standby.
- ExecuteMethod is for developers to play around. Any of the methods shown for getMethodTypes seen on the plugin web page report for the TV can be executed. The developer has to look at the ExecuteMethod outcome in the log file.
- Many TV calls have a context, so for example:
  - In 'Standby' you can execute any IR code with no errors.
  - In 'Standby' you cannot access the Volume or Mute levels; the display must be on.
  - In 'Standby' you can 'SetActiveApp' but a timeout error will occur while the TV starts up.

Usage

Variables you can examine, noting that all variables names are case sensitive:

The 'Connected' variable indicates if the TV is A/C powered on and contactable via the LAN. It can take up to 30 seconds (the polling period) before the variable indicates the correct state. Any code you write, can ensure if the unit is connected by checking the 'Connected' variable first.

Despite the 30 second delay above, most (not all) commands executed via the plugin, will update the status almost immediately.

The plugin's web page is useful for understanding general operation.

Examples:

```
Sending an IR code:
luup.call_action('urn:a-lurker-com:serviceId:Sony_Bravia_IP1', 'SendRemoteCode',
{RemoteCode = 'WakeUp'}, DeviceID)

Mute the TV:
luup.call_action('urn:a-lurker-com:serviceId:Sony_Bravia_IP1', 'SetMute', {Mute
= '1'}, DeviceID)

Launch an application:
luup.call_action('urn:a-lurker-com:serviceId:Sony_Bravia_IP1', 'SetActiveApp',
{Uri = 'com.sony.dtv.com.netflix.ninja.com.netflix.ninja.MainActivity'},
DeviceID)
```

Some commands take time to finalise eg 'Wakeup', so the user may need to use delays before sending the next command to the TV.

It's typically best to avoid the function sleep() as its usage may result in an untimely and unwanted Luup engine restart:

luup.sleep(msec) ← best not to use this approach

Use a sequence of delays with callback functions as one possible solution eg:

```
local function turnOnTV()
    ...
end

function selectInput()
    ...
end

function doSomethingElse()
    ...
end

function doFurtherAction()
    ...
end

Turn on the TV. It will take say 4 seconds for the TV to be ready
turnOnTV()

After 4 secs, the TV is now powered on and ready. Select an input
luup.call_delay("selectInput", 4, "")

After 5 secs, the input is selected; do something else
luup.call_delay("doSomethingElse", 4+1, "")

After 6 secs, the something else is complete; now do a further action
luup.call_delay("doFurtherAction", 4+1+1, "")
```

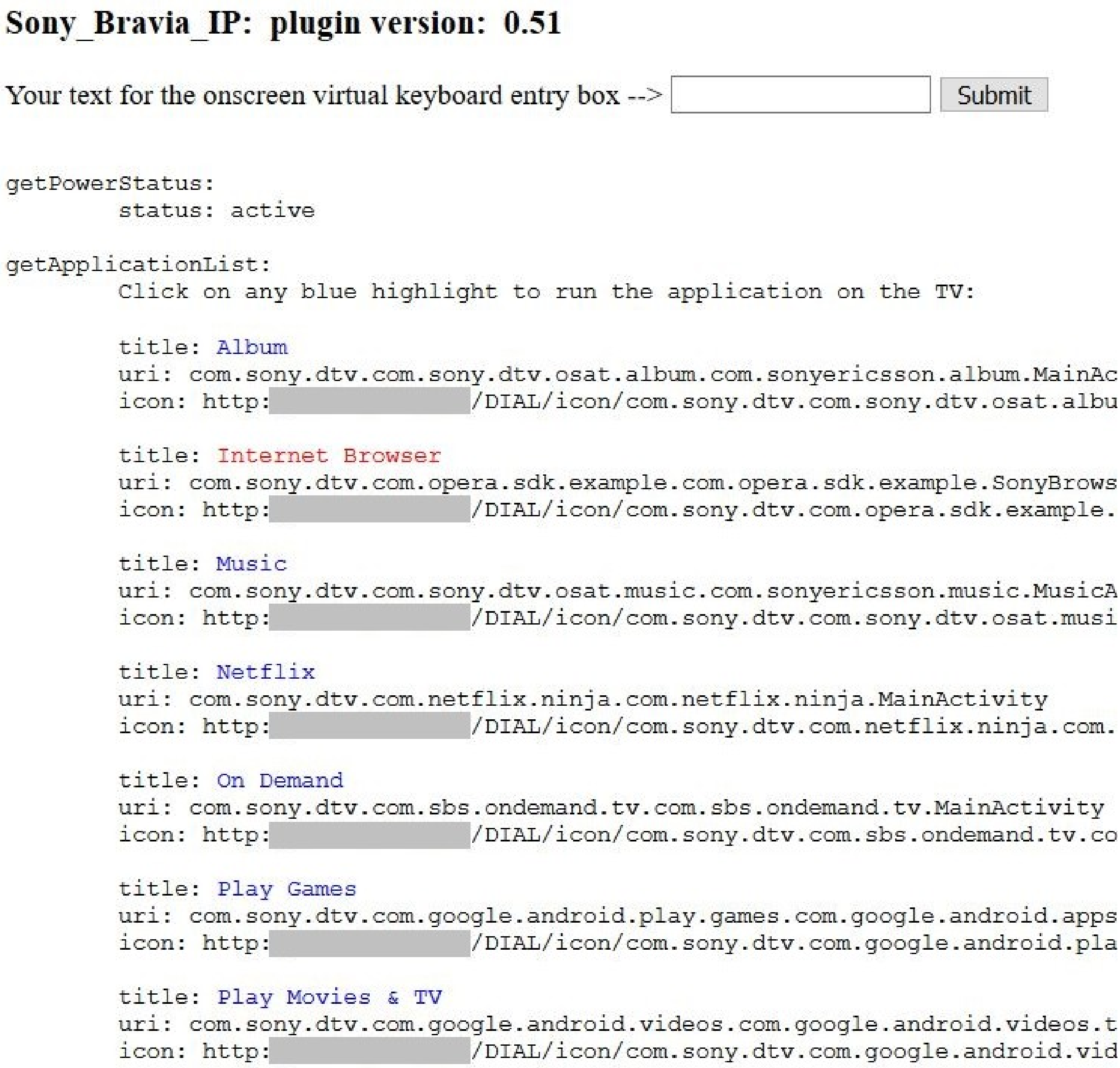
Debug

If DebugEnabled is set to '1' the debug info is enabled. Any debug commands will show up in the log file. Search on the string:

```
'Sony_Bravia_IP debug:'
```

DebugEnabled should be set to '0' in normal operation.

Section of the web page ie not the full list, showing some of the applications installed on the TV. You can invoke an application by clicking on its link:

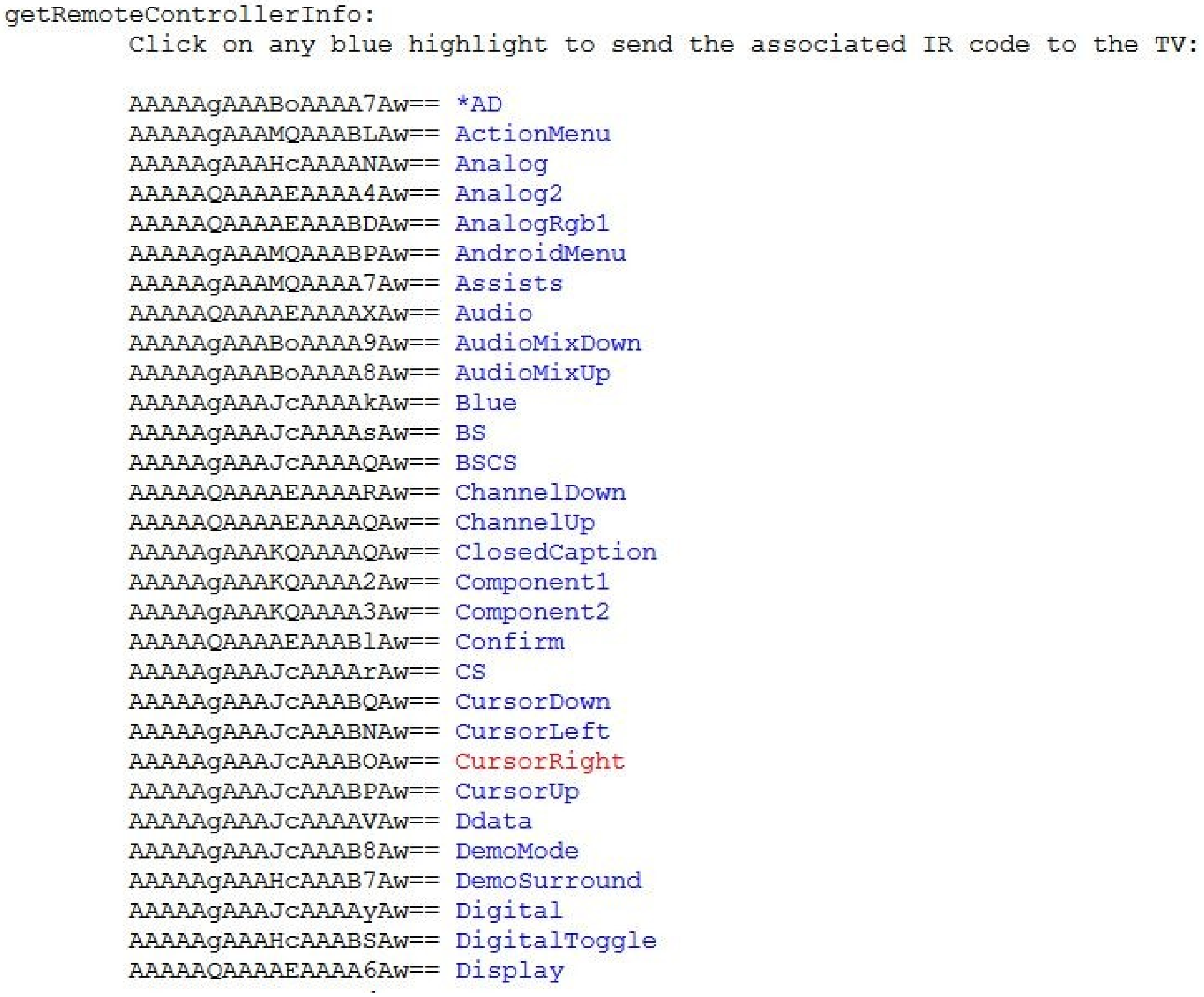


Sony_Bravia_IP: plugin version: 0.51

Your text for the onscreen virtual keyboard entry box -->

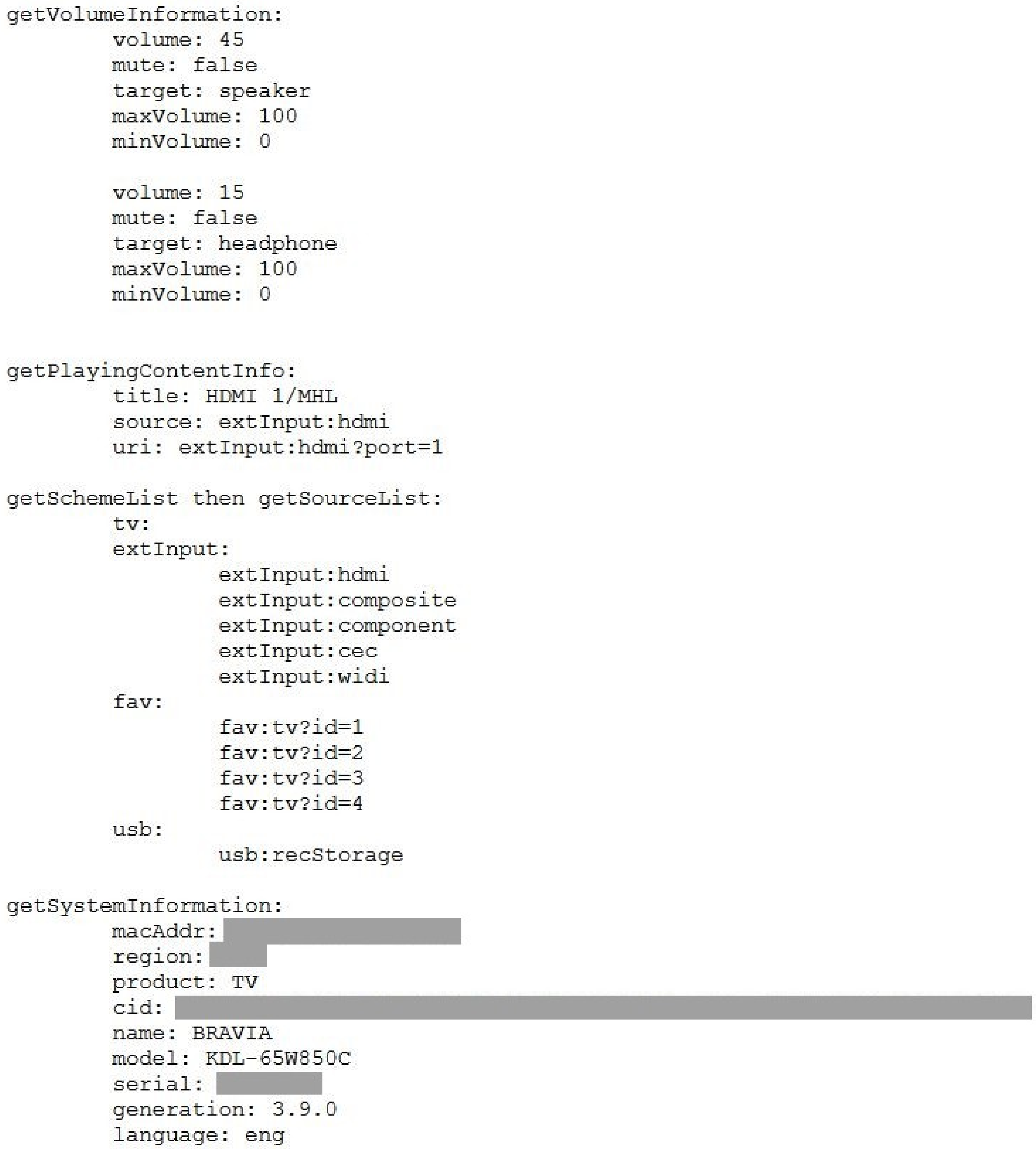
```
getPowerStatus:
        status: active

getApplicationList:
        Click on any blue highlight to run the application on the TV:

        title: Album
        uri: com.sony.dtv.com.sony.dtv.osat.album.com.sonyericsson.album.MainAc
        icon: http:[redacted]/DIAL/icon/com.sony.dtv.com.sony.dtv.osat.albu

        title: Internet Browser
        uri: com.sony.dtv.com.opera.sdk.example.com.opera.sdk.example.SonyBrows
        icon: http:[redacted]/DIAL/icon/com.sony.dtv.com.opera.sdk.example.

        title: Music
        uri: com.sony.dtv.com.sony.dtv.osat.music.com.sonyericsson.music.MusicA
        icon: http:[redacted]/DIAL/icon/com.sony.dtv.com.sony.dtv.osat.musi

        title: Netflix
        uri: com.sony.dtv.com.netflix.ninja.com.netflix.ninja.MainActivity
        icon: http:[redacted]/DIAL/icon/com.sony.dtv.com.netflix.ninja.com.

        title: On Demand
        uri: com.sony.dtv.com.sbs.ondemand.tv.com.sbs.ondemand.tv.MainActivity
        icon: http:[redacted]/DIAL/icon/com.sony.dtv.com.sbs.ondemand.tv.co

        title: Play Games
        uri: com.sony.dtv.com.google.android.play.games.com.google.android.apps
        icon: http:[redacted]/DIAL/icon/com.sony.dtv.com.google.android.pla

        title: Play Movies & TV
        uri: com.sony.dtv.com.google.android.videos.com.google.android.videos.t
        icon: http:[redacted]/DIAL/icon/com.sony.dtv.com.google.android.vid
```

Section of the web page ie not the full list, showing some of the IR codes available in the TV. You can send an IR code by clicking on its link:



```
getRemoteControllerInfo:
        Click on any blue highlight to send the associated IR code to the TV:

        AAAAAgAAABoAAAA7Aw== *AD
        AAAAAgAAAMQAAABLAw== ActionMenu
        AAAAAgAAAHcAAAANAw== Analog
        AAAAAQAAAAEAAAA4Aw== Analog2
        AAAAAQAAAAEAAABDAw== AnalogRgb1
        AAAAAgAAAMQAAABPAw== AndroidMenu
        AAAAAgAAAMQAAAA7Aw== Assists
        AAAAAQAAAAEAAAAXAw== Audio
        AAAAAgAAABoAAAA9Aw== AudioMixDown
        AAAAAgAAABoAAAA8Aw== AudioMixUp
        AAAAAgAAAJcAAAAkAw== Blue
        AAAAAgAAAJcAAAAsAw== BS
        AAAAAgAAAJcAAAAQAw== BSCS
        AAAAAQAAAAEAAAARAw== ChannelDown
        AAAAAQAAAAEAAAAQAw== ChannelUp
        AAAAAgAAAKQAAAAQAw== ClosedCaption
        AAAAAgAAAKQAAAA2Aw== Component1
        AAAAAgAAAKQAAAA3Aw== Component2
        AAAAAQAAAAEAAABlAw== Confirm
        AAAAAgAAAJcAAAArAw== CS
        AAAAAgAAAJcAAABQAw== CursorDown
        AAAAAgAAAJcAAABNAw== CursorLeft
        AAAAAgAAAJcAAABOAw== CursorRight
        AAAAAgAAAJcAAABPAw== CursorUp
        AAAAAgAAAJcAAAAVAw== Ddata
        AAAAAgAAAJcAAAB8Aw== DemoMode
        AAAAAgAAAHcAAAB7Aw== DemoSurround
        AAAAAgAAAJcAAAAyAw== Digital
        AAAAAgAAAHcAAABSAw== DigitalToggle
        AAAAAQAAAAEAAAA6Aw== Display
```

Section of the web page showing some of the TV's status, etc.



```
getVolumeInformation:
        volume: 45
        mute: false
        target: speaker
        maxVolume: 100
        minVolume: 0

        volume: 15
        mute: false
        target: headphone
        maxVolume: 100
        minVolume: 0


getPlayingContentInfo:
        title: HDMI 1/MHL
        source: extInput:hdmi
        uri: extInput:hdmi?port=1

getSchemeList then getSourceList:
        tv:
        extInput:
                extInput:hdmi
                extInput:composite
                extInput:component
                extInput:cec
                extInput:widi
        fav:
                fav:tv?id=1
                fav:tv?id=2
                fav:tv?id=3
                fav:tv?id=4
        usb:
                usb:recStorage

getSystemInformation:
        macAddr: [redacted]
        region: [redacted]
        product: TV
        cid: [redacted]
        name: BRAVIA
        model: KDL-65W850C
        serial: [redacted]
        generation: 3.9.0
        language: eng
```