User Manual - RDK Services on RPi

1. Introduction

RDK Services are the RDK components implemented as the Thunder Nano services. RDK services are the services that are developed based on the Thunder (WPE) Framework. Services each other or a particular service can be controlled through JSON RPC commands and it has a web-based controller UI.

The graphics UI services like WebKit Browser and so on are run on top of RDK Shell. RDK Shell is a native component that gives the Wayland composition with application management.

For the Integration of RDK Services on Generic RDK image, below are the important components

- RDK Shell
- WPE's Thunder Framework
- RDK Services (Thunder Nano services developed by Comcast)
- WPE Framework's UI (for Controller UI)
- WebKit Browser Plugin (A Thunder Nano service for WPE WebKit Browser)

Table of Contents

- 1. Introduction
- 2. Yocto build
- 3. User Interface
  - 3.1. App UI
    - 3.1.1. ResidentApp
    - 3.1.2. LightningApp
    - 3.1.3. Cobalt
    - 3.1.4. SearchandDiscoveryApp
    - 3.1.5. HTMLApp
    - 3.1.6. WebKitBrowser
  - 3.2. App Vs Instance
  - 3.3. Controller UI
    - 3.3.1. ResidentApp Control
- 4. RDK Services

2. Yocto build

The links are given here provide the instructions to

- setup host environment
- generate RDK Image for Media Client and Media Gateway through a Yocto 2.2 (Morty) based build framework
- the instruction to flash the images to a micro SD card

These instructions are made for the execution of the RDK media stack on the Raspberrypi 3 B+ Model.

- RDK-V (Raspberry Pi) Yocto Build Instructions - Morty

3. User Interface

User Interface involves graphics display on TV to show the app launched on the box and a controller UI that can be launched from the host machine's web browser to control all RDK services remotely.

3.1. App UI

SwitcherApp is a Lightning framework based JS application that launches on a WebKit Browser instance. The URL and other parameters of this WebKit Browser instance can be controller from the *ResidentApp* service on controller UI (from the Host machine's web browser).

SwitcherApp URL: https://px-apps.sys.comcast.net/lightning_apps/SwitcherApp/dist/index.html

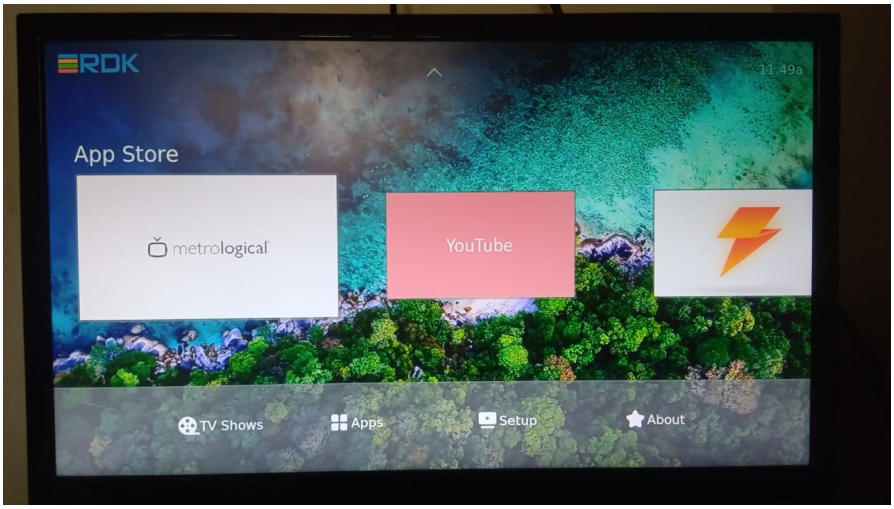
Note: App's can be navigated through a Keyboard or an IR (especially Xfinity) remote connected to the RPi

3.1.1. ResidentApp

Resident App is a WebKit browser instance where it loads the App resides in the box. By default, it loads with Switcher App URL. The switcher App loads different WebKit browser engine for various apps

Switcher APP UI

Swither App launches through *ResidentApp* (WebKit Browser) Instance



Note: Numerical `1` key is used to come out of any apps on this URL

3.1.2. LightningApp

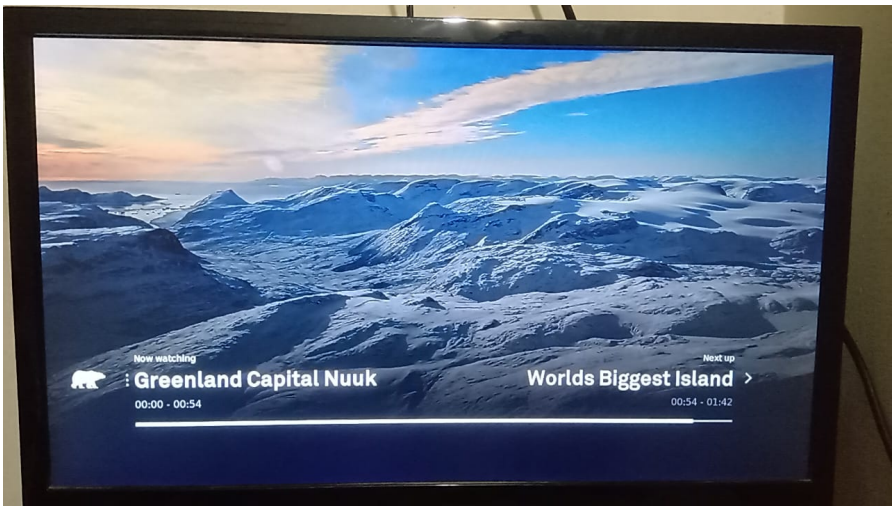
LightningApp is another WebKit browser instance specifically for apps designed through the Lightning framework.

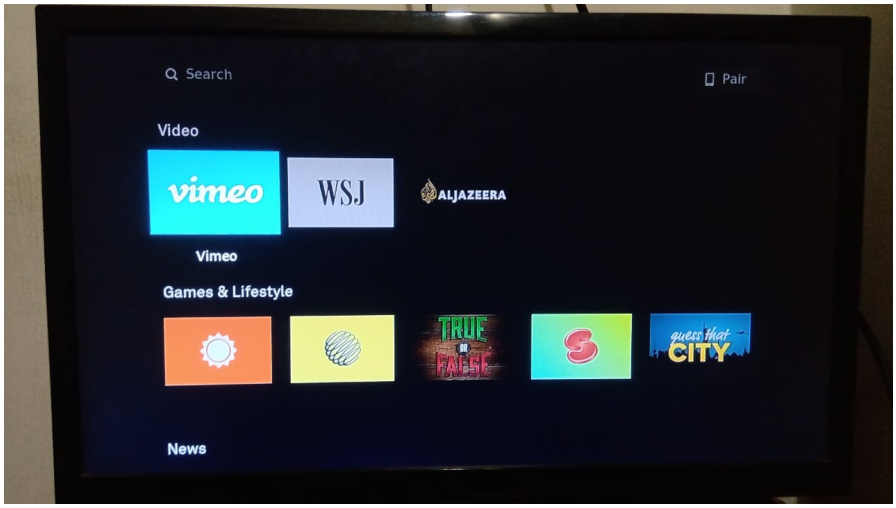
Metrological's App Functionality

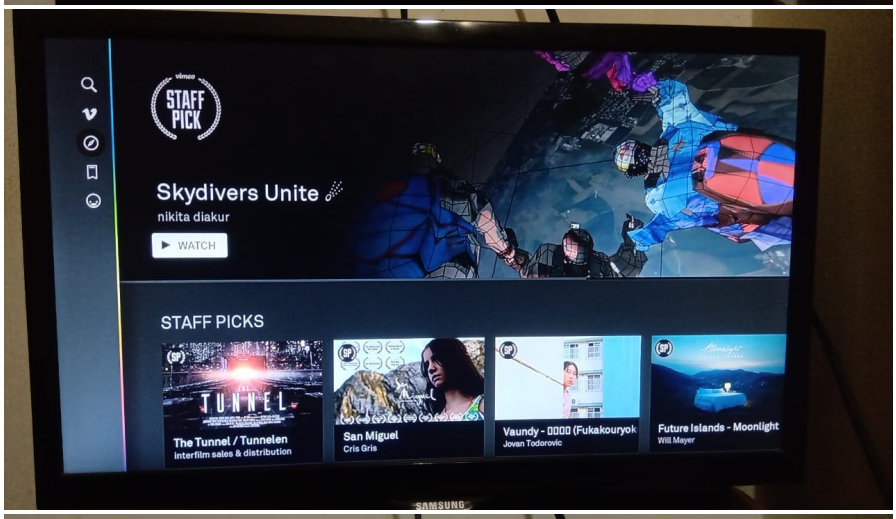
Metrological's App launches on *LightningApp* (WebKit Browser) Instance. Though it available on Switcher App it launches in a different browser instance.

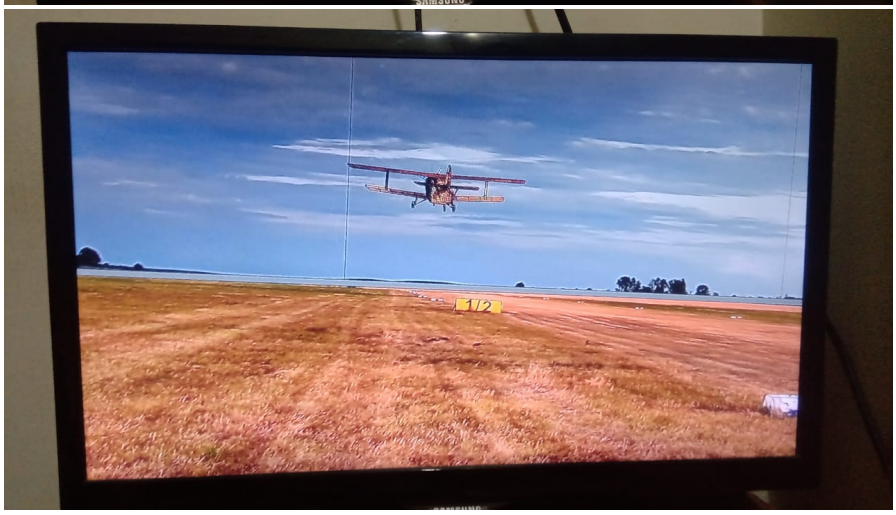
Metrological's App contains

- live video playback
- Vimeo (VOD playback) and other video applications
- Other Applications









3.1.3. Cobalt

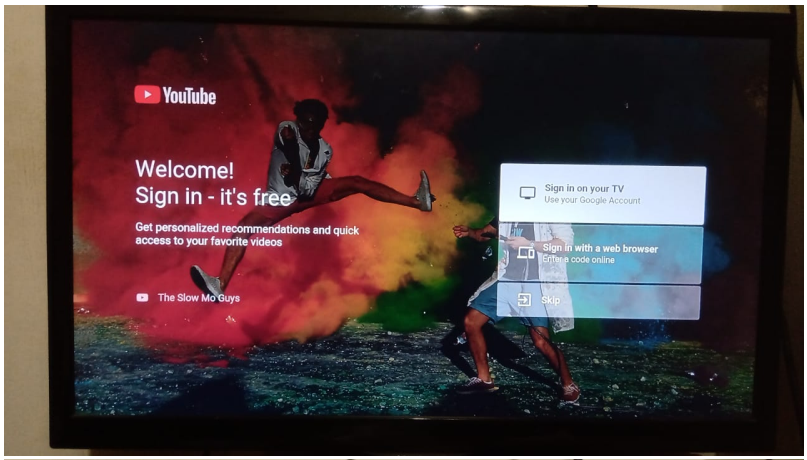
Cobalt is an RDK service that gives a lightweight HTML5/CSS/JS based lightweight application container that is mainly designed for a rich animation environment. This service is mainly used for Youtube TV.

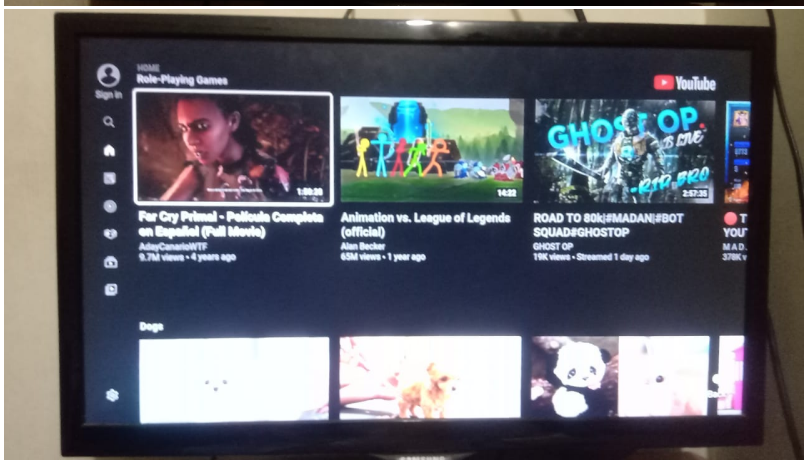
Youtube TV

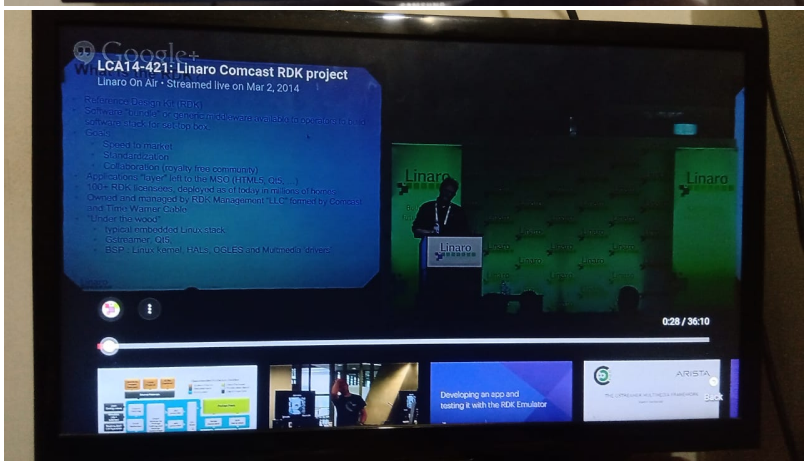
Youtube from Switcher App launches through Cobalt.

Youtube can be launched from

- Switcher App
- Cobalt Service

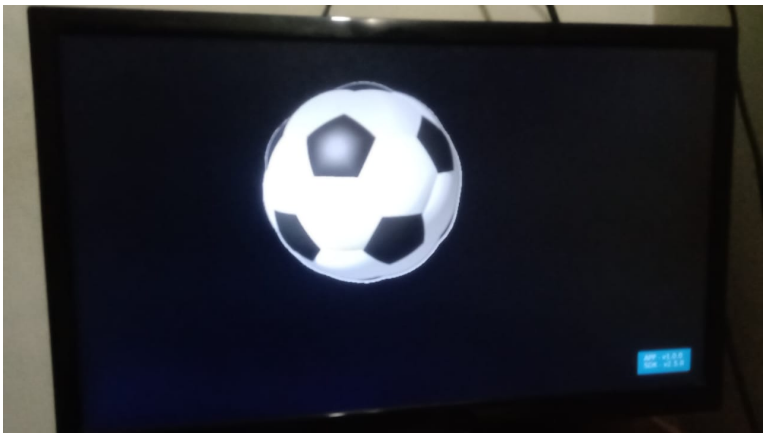






Bouncing Ball

Bouncing ball is a Lightning framework based Graphics rendering App) that comes as one of an app in the Switcher App. On launching the bouncing ball, a *LightningApp* (WebKit browser) instance will be created.



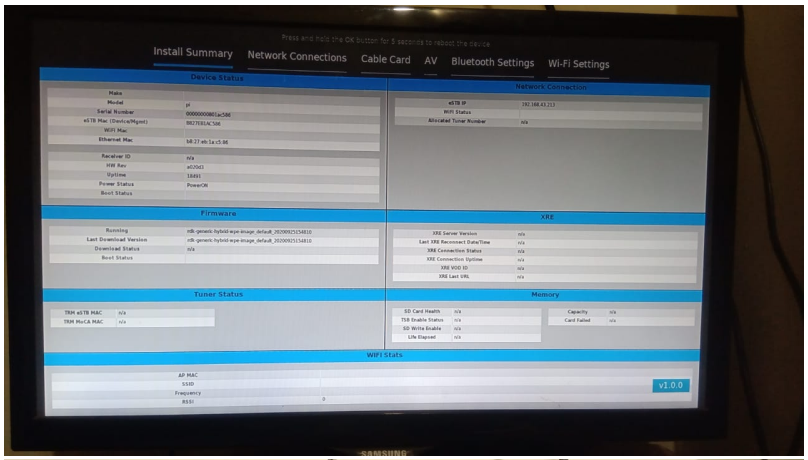
3.1.4. SearchandDiscoveryApp

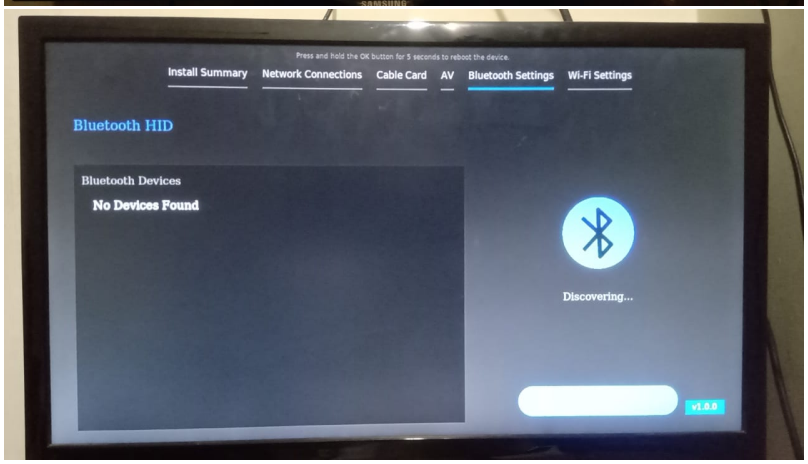
SearchandDiscoveryApp is another WebKit Browser instance mainly used for setup functionality

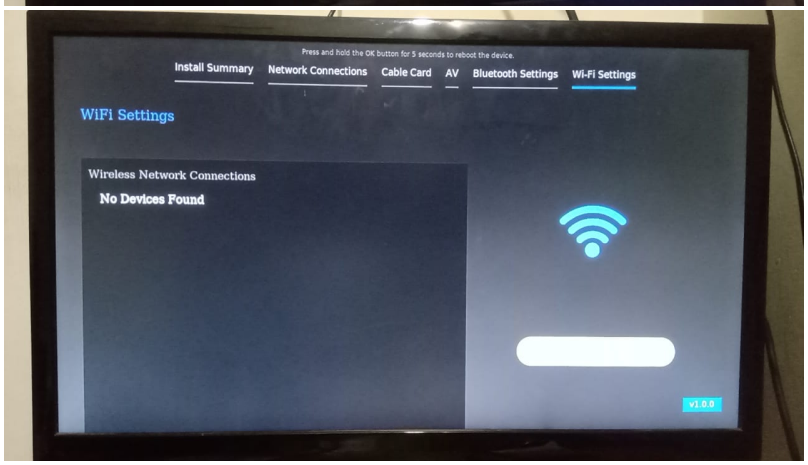
Setup

Setup configuration and settings which is from Switcher App that contains

- Install summary - Info page to display the device and image details
- Network Connection - Info page to display network interfaces details
- Cable Card - Info page about the Media supported
- Bluetooth Settings - Settings page to set up a Bluetooth Audio device connection
- WiFi Settings - Settings page to set up a WiFi network connection







3.1.5. HTMLApp

HTMLApp is another WebKit browser instance or service that mainly used for HTML-based applications.

3.1.6. WebKitBrowser

WebKit Browser is a base browser service where any external URL can be loaded.

3.2. **App Vs Instance**

Below given apps will be loaded on the corresponding UI browser service

Metrological App - Lightning App

Switcher App URL - Resident App

Youtube - Cobalt

Bouncing Ball - Lightning App

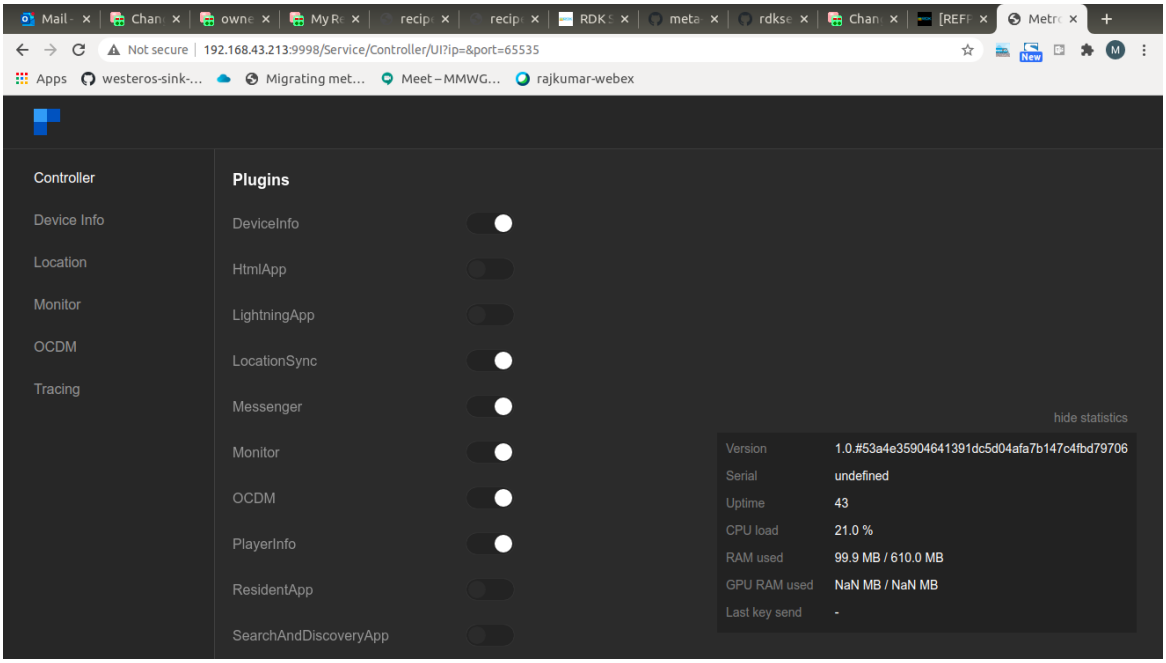
Setup menu on Switcher App - SearchAndDiscoveryApp

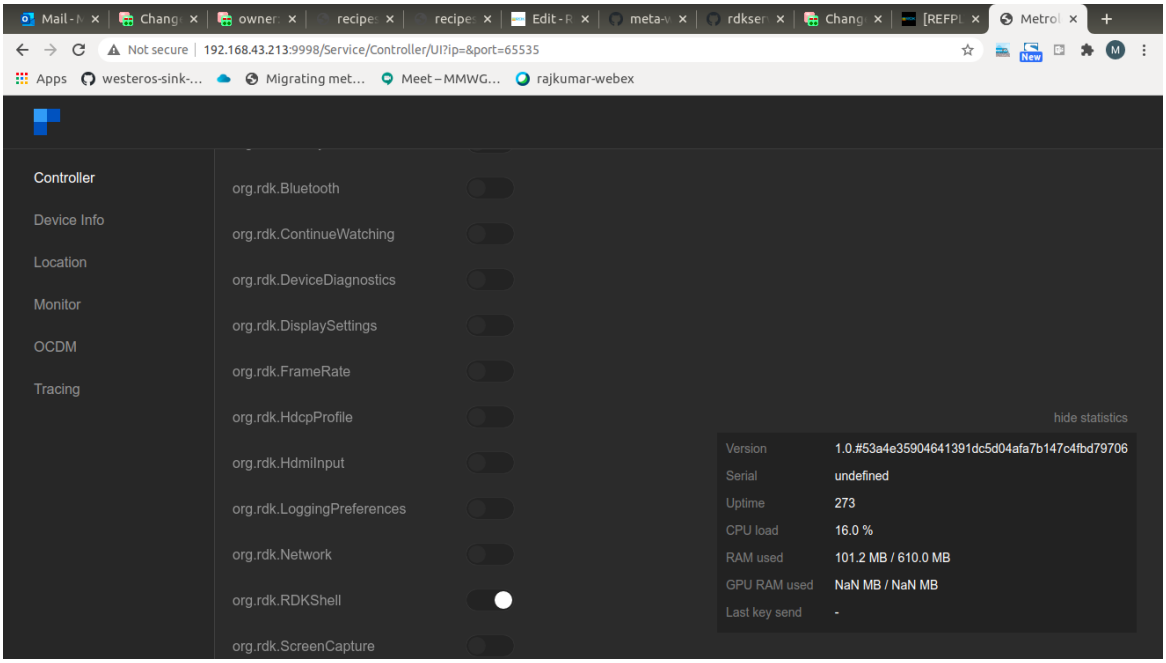
3.3. Controller UI

Controller UI is a web UI that can be launched from a host machine's (machine under the same network where RPi resides) browser. This UI can be loaded with the RPi box's IP address with port number 9998.
URL: `http://<Rpi's IP address>:9998`

Controller: List of plugins

In the *Controller* tab, any plugins can be enabled or disabled. On loading the controller UI initially, it loads the *Controller* tab with all possible services with the enable or disable status.

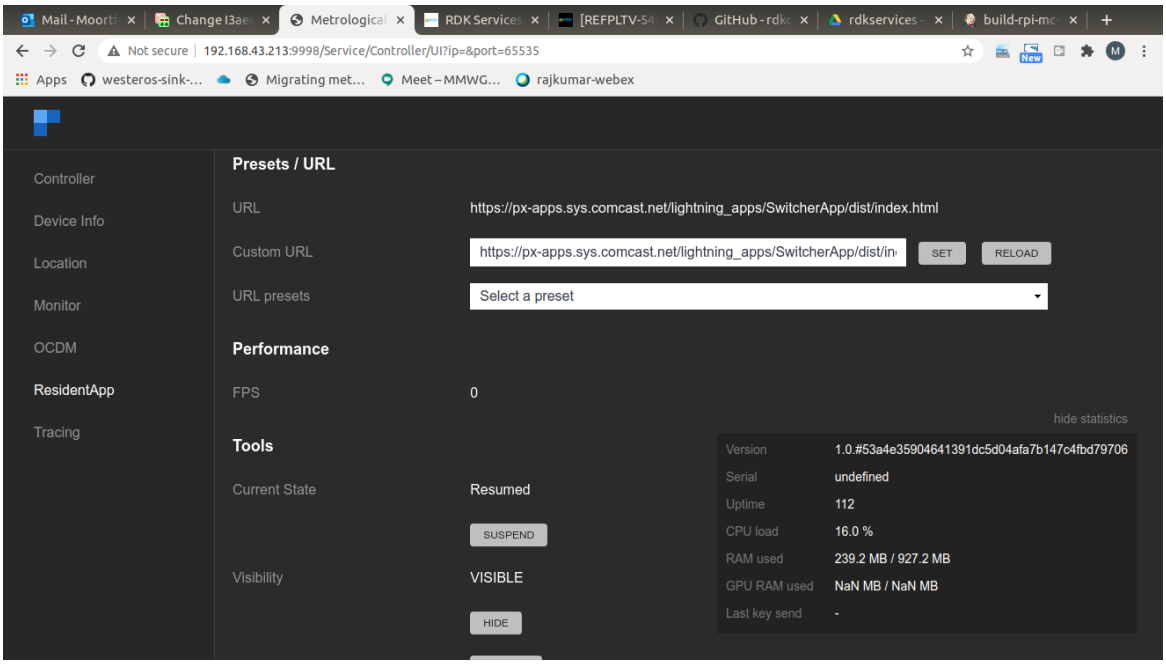




3.3.1. ResidentApp Control

As this is a WebKit browser instance, preset and custom URL can be loaded from the controller UI.

ResidentApp Control



4. RDK Services

Each RDK Service can be validated through JSON RPC Commands through HTTP. It has a request and response in JSON format. Some RDK Services which have graphics UI can be validated through the controller UI as well.

As the JSON RPC commands to control the RDK service through HTTP, requests can be sent from the target and from the remote (a host connected on the same network) machines.

- On target - Use box's IP or the local host's IP
- On Remote - Use only the box's IP

Controller

Any service can be enabled or disabled also through JSONRPC commands.

Function	Request	Response	Remarks
activate	`curl http://localhost:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{ "jsonrpc":"2.0", "id":3, "method":"Controller.1.activate", "params":{"callsign":"org.rdk.RDKShell"} }'`	`{"jsonrpc":"2.0","id":3,"result":{"success":true}}`	callsign argument can accept all plugins Note: the argument is case sensitive
deactivate	`curl http://localhost:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{ "jsonrpc":"2.0", "id":3, "method":"Controller.1.deactivate", "params":{"callsign":"org.rdk.RDKShell"} }'`	`{"jsonrpc":"2.0","id":3,"result":{"success":true}}`	
environment	`curl http://localhost:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{ "jsonrpc":"2.0", "id":3, "method":"Controller.1.environment@XDG_RUNTIME_DIR"}'`	`{"jsonrpc":"2.0","id":3,"result":"/tmp"}`	

org.rdk.RDKShell

RDK Shell as an RDK service enables by default which controls the management of Composition, Layout, Z-Order, and Key handling. Below JSON RPC commands are used to validate the RDK Shell's service functionality.

Prerequisite: This service has to be enabled from the controller UI to proceed with the execution of the below APIs

Function	Request	Response	Remarks
get Screen Resolution	`curl 'http://127.0.0.1:9998/jsonrpc' -d '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.RDKShell.1.getScreenResolution", "params":{ }}'`	`{"jsonrpc":"2.0", "id":3, "result": {"w": 1920, "h": 1080} }`	

set Screen Resolution	curl 'http://127.0.0.1:9998/jsonrpc' -d '{"jsonrpc":"2.0", "id":3, "method": "org.rdk.RDKShell.1.setScreenResolution", "params":{ "w": 1920, "h": 1080 }}'	{"jsonrpc":"2.0", "id":3, "result": {} }	
launch Application	curl 'http://127.0.0.1:9998/jsonrpc' -d '{"jsonrpc":"2.0", "id": 3, "method":"org.rdk.RDKShell.1.launch", "params": {"callsign":"HtmlApp", "type":"application", "visible":true, "uri":"http://www.goggle.com"}}'	{"jsonrpc":"2.0", "id":3, "result": {} }	
suspend Application	curl 'http://127.0.0.1:9998/jsonrpc' -d '{"jsonrpc":"2.0", "id":3, "method": "org.rdk.RDKShell.1.suspend", "params":{"client": "Cobalt"}}'	{"jsonrpc":"2.0", "id":3, "result": {} }	
resume Application	curl 'http://127.0.0.1:9998/jsonrpc' -d '{"jsonrpc":"2.0", "id":3, "method": "org.rdk.RDKShell.1.resume", "params":{"client": "Cobalt"}}'	{"jsonrpc":"2.0", "id":3, "result": {} }	
get Clients	curl 'http://127.0.0.1:9998/jsonrpc' -d '{"jsonrpc":"2.0", "id":3, "method": "org.rdk.RDKShell.1.getClients", "params":{ }}'	{"jsonrpc":"2.0", "id":3, "result": { "clients": ["ResidentApp", "org.rdk.RDKBrowser2"]} }	The below functionality will be succeeded only to the clients listed in this response
get Z-Order	curl 'http://127.0.0.1:9998/jsonrpc' -d '{"jsonrpc":"2.0", "id":3, "method": "org.rdk.RDKShell.1.getZOrder", "params":{ }}'	{"jsonrpc":"2.0", "id":3, "result": { "clients": ["ResidentApp", "org.rdk.RDKBrowser2"]} }	
Move to Front	curl 'http://127.0.0.1:9998/jsonrpc' -d '{"jsonrpc":"2.0", "id":3, "method": "org.rdk.RDKShell.1.moveToFront", "params":{ "client": "ResidentApp" }}'	{"jsonrpc":"2.0", "id":3, "result": {} }	*client* argument should be the exact name of the graphics UI service
Move to Back	curl 'http://127.0.0.1:9998/jsonrpc' -d '{"jsonrpc":"2.0", "id":3, "method": "org.rdk.RDKShell.1.moveToBack", "params":{ "client": "ResidentApp" }}'	{"jsonrpc":"2.0", "id":3, "result": {} }	
move Behind	curl 'http://127.0.0.1:9998/jsonrpc' -d '{"jsonrpc":"2.0", "id":3, "method": "org.rdk.RDKShell.1.moveBehind", "params":{ "client": "WebKitBrowser", "target": "ResidentApp"}}'	{"jsonrpc":"2.0", "id":3, "result": {} }	
set Focus	curl 'http://127.0.0.1:9998/jsonrpc' -d '{"jsonrpc":"2.0", "id":3, "method": "org.rdk.RDKShell.1.setFocus", "params":{ "client": "WebKitBrowser" }}'	{"jsonrpc":"2.0", "id":3, "result": {} }	
get Bounds	curl 'http://127.0.0.1:9998/jsonrpc' -d '{"jsonrpc":"2.0", "id":3, "method": "org.rdk.RDKShell.1.getBounds", "params":{ "client": "ResidentApp"}}'	{"jsonrpc":"2.0", "id":3, "result": { "x": 0, "y": 0, "w": 600, "h": 400} }	
set Bounds	curl 'http://127.0.0.1:9998/jsonrpc' -d '{"jsonrpc":"2.0", "id":3, "method": "org.rdk.RDKShell.1.setBounds", "params":{ "client": "WebKitBrowser", "x": 0, "y": 0, "w": 600, "h": 400 }}'	{"jsonrpc":"2.0", "id":3, "result": {} }	
get Visibility	curl 'http://127.0.0.1:9998/jsonrpc' -d '{"jsonrpc":"2.0", "id":3, "method": "org.rdk.RDKShell.1.getVisibility", "params":{ "client": "ResidentApp" }}'	{"jsonrpc":"2.0", "id":3, "result": {"visible": true} }	
set Visibility	curl 'http://127.0.0.1:9998/jsonrpc' -d ' {"jsonrpc":"2.0", "id":3, "method":"org.rdk.RDKShell.1.setVisibility", "params":{ "client": "ResidentApp", "visible": true }}'	{"jsonrpc":"2.0", "id":3, "result": {} }	
get Opacity	curl 'http://127.0.0.1:9998/jsonrpc' -d '{"jsonrpc":"2.0", "id":3, "method": "org.rdk.RDKShell.1.getOpacity", "params":{ "client": "ResidentApp"}}'	{"jsonrpc":"2.0", "id":3, "result": { "opacity" : 100} }	
set Opacity	curl 'http://127.0.0.1:9998/jsonrpc' -d '{"jsonrpc":"2.0", "id":3, "method": "org.rdk.RDKShell.1.setOpacity", "params":{ "client": "ResidentApp", "opacity": 100}}'	{"jsonrpc":"2.0", "id":3, "result": {} }	
get Scale	curl 'http://127.0.0.1:9998/jsonrpc' -d '{"jsonrpc":"2.0", "id":3, "method": "org.rdk.RDKShell.1.getScale", "params":{"client": "ResidentApp"}}'	{"jsonrpc":"2.0", "id":3, "result": {"sx":1.5, "sy":1.5} }	
set Scale	curl 'http://127.0.0.1:9998/jsonrpc' -d '{"jsonrpc":"2.0", "id":3, "method": "org.rdk.RDKShell.1.setScale", "params":{"client": "ResidentApp", "sx":1.5, "sy":1.5}}'	{"jsonrpc":"2.0", "id":3, "result": {} }	
kill	curl 'http://127.0.0.1:9998/jsonrpc' -d '{"jsonrpc":"2.0", "id":3, "method": "org.rdk.RDKShell.1.kill", "params":{ "client": "WebKitBrowser" }}'	{"jsonrpc":"2.0", "id":3, "result": {} }	
generate Key	curl 'http://127.0.0.1:9998/jsonrpc' -d '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.RDKShell.1.generateKey", "params":{"keys":[{"keyCode": 49,"modifiers": [],"delay":1.0}{"keyCode": 50,"modifiers": [],"delay": 2.0}]}}'	{"jsonrpc":"2.0", "id":3, "result": {} }	

Note: The above functions are referred from the RDK Service's RDK Shell wiki page

org.rdk.Network

Prerequisite: This service has to be enabled from the controller UI to proceed with the execution of the below APIs

Function	Request	Response	Remarks

get Interfaces	curl http://localhost:9998/jsonrpc -d '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.Network.1.getInterfaces", "params":{}}'	{"jsonrpc":"2.0", "id":3, "result": {"interfaces": [{ "interface":"WIFI", "macAddress":"AA:AA:AA:AA:AA:AA", "enabled":true, "connected":true }, { "interface":"ETHERNET", "macAddress":"AA:AA:AA:AA:AA:AD", "enabled":false, "connected":false }] } }	
is Interface Enabled	curl http://localhost:9998/jsonrpc -d '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.Network.1.isInterfaceEnabled", "params":{"interface":"WIFI"}}'	{"jsonrpc":"2.0", "id":3, "result": {"enabled":true} }	
Enable Interface	curl http://localhost:9998/jsonrpc -d '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.Network.1.setInterfaceEnabled", "params":{"interface":"WIFI", "enabled":true, "persist":true}}'	{"jsonrpc":"2.0", "id":3, "result": 0}	
set Default Interface	curl http://localhost:9998/jsonrpc -d '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.Network.1.setDefaultInterface", "params":{ "interface":"WIFI", "persist":true}}'	{"jsonrpc":"2.0", "id":3, "result": {}}	
get Default Interface	curl http://localhost:9998/jsonrpc -d '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.Network.1.getDefaultInterface", "params":{}}'	{"jsonrpc":"2.0", "id":3, "result": {"interface":"WIFI"}}	
get IP for default interface	curl http://localhost:9998/jsonrpc -d '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.Network.1.getStbIp", "params":{}}'	{"jsonrpc":"2.0","id":3,"result": {"ip":"192.168.xx.yyy","success": true}}	
ping	curl http://localhost:9998/jsonrpc -d '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.Network.1.ping", "params":{ "endpoint": "45.57.221.20", "packets": 10 }}'	{"jsonrpc":"2.0", "id":3, "result":{ "target" : "173.194.123.7", "success" : true, "packetsTransmitted" : 10, "packetsReceived" : 10, "packetLoss" : "0.0", "tripMin" : "61.264", "tripAvg" : "130.397", "tripMax" : "230.832", "tripStdDev" : "80.919", "error" : null }}	
get Named Endpoints	curl http://localhost:9998/jsonrpc -d '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.Network.1.getNamedEndpoints", "params":{}}'	{"jsonrpc":"2.0", "id":3, "result":{"endpoints":["CMTS"]}}	
ping to named Endpoint	curl http://localhost:9998/jsonrpc -d '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.Network.1.pingNamedEndpoint", "params":{ "endpointName": "CMTS", "packets": 10 }}'	{"jsonrpc":"2.0", "id":3, "result":{ "target" : "173.194.123.7", "success" : true, "packetsTransmitted" : 10, "packetsReceived" : 10, "packetLoss" : "0.0", "tripMin" : "61.264", "tripAvg" : "130.397", "tripMax" : "230.832", "tripStdDev" : "80.919", "error" : null }}	
trace	curl http://localhost:9998/jsonrpc -d '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.Network.1.trace", "params":{ "endpoint": "45.57.221.20", "packets": 10 }}'	{"jsonrpc":"2.0", "id":3, "result":{ "target" : "173.194.123.7", "success" : true, "error" : null, "results" : "<<<traceroute command results>>>" }}	
trace a named Endpoint	curl http://localhost:9998/jsonrpc -d '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.Network.1.traceNamedEndpoint", "params":{ "guid": "iasoviubejcv", "endpointName": "CMTS", "packets": 10 }}'	{"jsonrpc":"2.0", "id":3, "result":{ "target" : "173.194.123.7", "success" : true, "error" : null, "results" : "<<<traceroute command results>>>" }}	
get IP Settings	curl http://localhost:9998/jsonrpc -d '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.Network.1.getIPSettings", "params":{ "interface": "wlan0" }}'	{"jsonrpc":"2.0", "id":3, "result":{ "interface": "wifi", "ipversion": "IPv4 "autoconfig": true, "ipaddr": “192.168.1.101", "netmask": "255.255.255.0", "gateway": "192.168.1.1", "primarydns": "192.168.1.1", "secondarydns": "" } }	

set IP Settings	curl http://localhost:9998/jsonrpc -d '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.Network.1.setIPSettings", "params":{ "interface": "wifi", "ipversion": "IPv4", "autoconfig": true, "ipaddr": "192.168.1.101", "netmask": "255.255.255.0", "gateway": "192.168.1.1", "primarydns": "192.168.1.1", "secondarydns": "" }}'	{"jsonrpc":"2.0", "id":3, "result":{ "supported": true, "success": true } }	

Note: The functions are referred from the RDK Service's Network wiki page

org.rdk.Wifi

Prerequisite: This service has to be enabled from the controller UI to proceed with the execution of the below APIs

Function	Request	Response	Remarks
get Current State	curl http://127.0.0.1:9998/jsonrpc -d '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.Wifi.1.getCurrentState", "params":{}}'	{"jsonrpc":"2.0", "id":3, "result": {"state":2,"success":true}}	
start Scan	curl http://127.0.0.1:9998/jsonrpc -d '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.Wifi.1.startScan", "params":{"incremental":false,"ssid":"","frequency":""}}'	{"jsonrpc":"2.0", "id":3, "result":{"success":true}}	incremental can be set to true to get each SSID in different events (mostly used from GUI)
stop scan	curl http://127.0.0.1:9998/jsonrpc -d '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.Wifi.1.stopScan", "params":{}}'	{"jsonrpc":"2.0", "id":3, "result":{"success":true}}	
set Enabled	curl http://127.0.0.1:9998/jsonrpc -d '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.Wifi.1.setEnabled", "params":{"enable":true}}'	{"jsonrpc":"2.0", "id":3, "result":{"success":true}}	
connect	curl http://127.0.0.1:9998/jsonrpc -d '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.Wifi.1.connect", "params":{"ssid":"123412341234", "passphrase": "foobar", "securityMode": 3}}'	{"jsonrpc":"2.0", "id":3, "result":{"success":true}}	
get connected SSID	curl http://127.0.0.1:9998/jsonrpc -d '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.Wifi.1.getConnectedSSID", "params":{}}'	{"jsonrpc":"2.0", "id":3, "result":{"ssid":"TP-LINK 1234","bssid":"ec:08:6b:ce:09:20", "rate":"144.000000","noise":"-121.000000", "security":"5","signalStrength":"-27.000000","frequency":"2.442000","success":true}	
disconnect	curl http://127.0.0.1:9998/jsonrpc -d '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.Wifi.1.disconnect", "params":{}}'	{"jsonrpc":"2.0", "id":3, "result":{"result":"","success":true}}	
initiate pairing through WPS	curl http://127.0.0.1:9998/jsonrpc -d '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.Wifi.1.initiateWPSPairing"}'	{"jsonrpc":"2.0", "id":3, "result":{"result":"","success":true}}	
cancel WPS pairing	curl http://127.0.0.1:9998/jsonrpc -d '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.Wifi.1.cancelWPSPairing", "params":{}}'	{"jsonrpc":"2.0", "id":3, "result":{"result":"","success":true}}	
save SSID	curl http://127.0.0.1:9998/jsonrpc -d '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.Wifi.1.saveSSID", "params":{"ssid": "123412341234", "passphrase": "foobar", "securityMode": 3}}'	{"jsonrpc":"2.0", "id":3, "result":{"result":0,"success":true}}	
is Paired	curl http://127.0.0.1:9998/jsonrpc -d '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.Wifi.1.isPaired", "params":{}}'	{"jsonrpc":"2.0", "id":3, "result":{"result":0,"success":true}}	
get Paired SSID	curl http://127.0.0.1:9998/jsonrpc -d '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.Wifi.1.getPairedSSID", "params":{}}'	{"jsonrpc":"2.0", "id":3, "result":{"ssid":"123412341234","success":true}}}	
get Paired SSID Info	curl http://127.0.0.1:9998/jsonrpc -d '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.Wifi.1.getPairedSSIDInfo", "params":{}}'	{"jsonrpc":"2.0", "id":3, "result":{"ssid":"123412341234", "bssid": "ff:ff:ff:ff:ff:ff,"success":true}}	not applicable for RPi as *nets rvmgr* has no 'lost & found' package
clear SSID	curl http://127.0.0.1:9998/jsonrpc -d '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.Wifi.1.clearSSID", "params":{}}'	{"jsonrpc":"2.0", "id":3, "result":{"result":0,"success":true}}	
set Signal Threshold change enabled	curl http://127.0.0.1:9998/jsonrpc -d '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.Wifi.1.setSignalThresholdChangeEnabled", "params":{"enabled": true, "interval": 2000}}'	{"jsonrpc":"2.0", "id":3, "result":{"success":true}}	
is Signal Threshold change enabled	curl http://127.0.0.1:9998/jsonrpc -d '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.Wifi.1.isSignalThresholdChangeEnabled", "params":{}}'	{"jsonrpc":"2.0", "id":3, "result":{"result":1,"success":true}}	

Note: The functions are referred from the RDK Service's Wifi wiki page

org.rdk.Bluetooth

Prerequisite: This service has to be enabled from the controller UI to proceed with the execution of the below APIs

Function	Request	Response	Remarks
enable	curl --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":"3", "method":"org.rdk.Bluetooth.1.enable"}' http://127.0.0.1:9998/jsonrpc	{"jsonrpc":"2.0","id":3,"result":{"success":true}}	

disable	curl --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":"3", "method":"org.rdk.Bluetooth.1.disable"}' http://127.0.0.1:9998/jsonrpc	{"jsonrpc":"2.0","id":3,"result":{"success":true}}	
get Device Name	curl --header "Content-Type: application/json" --request POST --data {"jsonrpc":"2.0", "id":"3", "method":"org.rdk.Bluetooth.1.getName"}' http://127.0.0.1:9998/jsonrpc	{"jsonrpc":"2.0","id":3,"result":{"name":"Xfinity Bluetooth Audio","success":true}}	
set Device Name	curl --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":"3", "method":"org.rdk.Bluetooth.1.setName", "params":{"name": "RDK Bluetooth Device"}}' http://127.0.0.1:9998/jsonrpc	{"jsonrpc":"2.0","id":3,"result":{"success":true}}	
is Discoverable	curl --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":"3", "method":"org.rdk.Bluetooth.1.isDiscoverable"}' http://127.0.0.1:9998/jsonrpc	{"jsonrpc":"2.0","id":3,"result":{"discoverable":false,"success":true}}	
set Discoverable	curl --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":"3", "method":"org.rdk.Bluetooth.1.setDiscoverable", "params":{"discoverable":true, "timeout":10}}' http://127.0.0.1:9998/jsonrpc	{"jsonrpc":"2.0","id":3,"result":{"success":true}}	
start Scan	curl --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":"3", "method":"org.rdk.Bluetooth.1.startScan", "params": {"timeout": "5", "profile": "DEFAULT"}}' http://127.0.0.1:9998/jsonrpc	{"jsonrpc":"2.0","id":3,"result":{"status":"AVAILABLE","success":true}}	
stop scan	curl --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":"3", "method":"org.rdk.Bluetooth.1.stopScan"}' http://127.0.0.1:9998/jsonrpc	{"jsonrpc":"2.0","id":3,"result":{"success":true}}	
get Discovered Devices	curl --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":"3", "method":"org.rdk.Bluetooth.1.getDiscoveredDevices"}' http://127.0.0.1:9998/jsonrpc	{ "jsonrpc": "2.0", "id": 3, "result": { "discoveredDevices": [{ "deviceID": "61579454946360", "name": "[TV] UE32J5530", "deviceType": "TV", "connected": false, "paired": false }], "success": true } }	
pair	curl --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":"3", "method":"org.rdk.Bluetooth.1.pair", "params": {"deviceID": "256168644324480"}}' http://127.0.0.1:9998/jsonrpc	{"jsonrpc":"2.0","id":3,"result":{"success":true}}	
get Paired Devices	curl --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":"3", "method":"org.rdk.Bluetooth.1.getPairedDevices"}' http://127.0.0.1:9998/jsonrpc	{ "jsonrpc": "2.0", "id": 3, "result": { "pairedDevices": [{ "deviceID": "256168644324480", "name": "Eleven", "deviceType": "SMARTPHONE", "connected": true }, { "deviceID": "26499258260618", "name": "Little Big", "deviceType": "SMARTPHONE", "connected": false }], "success": true } }	
connect	curl --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":"3", "method":"org.rdk.Bluetooth.1.connect", "params": {"deviceID": "256168644324480", "deviceType": "SMARTPHONE", "profile": "SMARTPHONE"}}' http://127.0.0.1:9998/jsonrpc	{"jsonrpc":"2.0","id":3,"result":{"success":true}}	

get Connected Devices	curl --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":"3", "method":"org.rdk.Bluetooth.1.getConnectedDevices"}' http://127.0.0.1:9998/jsonrpc	{
		  "jsonrpc": "2.0",
		  "id": 3,
		  "result": {
		  "connectedDevices": [
		  {
		  "deviceID": "256168644324480",
		  "name": "Eleven",
		  "deviceType": "SMARTPHONE",
		  "activeState": "0"
		  }
		  ],
		  "success": true
		  }
		}	

unpair	curl --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":"3", "method":"org.rdk.Bluetooth.1.unpair", "params": {"deviceID": "256168644324480"}}' http://127.0.0.1:9998/jsonrpc	{"jsonrpc":"2.0","id":3,"result":{"success":true}}	
disconnect	curl --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":"3", "method":"org.rdk.Bluetooth.1.disconnect", "params": {"deviceID": "256168644324480"}}' http://127.0.0.1:9998/jsonrpc	{"jsonrpc":"2.0","id":3,"result":{"success":true}}	
set Audio Stream	curl http://127.0.0.1:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":"3", "method":"org.rdk.Bluetooth.1.setAudioStream", "params": {"deviceID": "256168644324480", "audioStreamName": "PRIMARY"}}'	{"jsonrpc":"2.0","id":3,"result":{"success":true}}	

get Device Info	curl http://127.0.0.1:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":"3", "method":"org.rdk.Bluetooth.1.getDeviceInfo", "params":{"deviceID":"256168644324480"}}'	{
		  "jsonrpc": "2.0",
		  "id": 3,
		  "result": {
		  "deviceInfo": {
		  "deviceID": "256168644324480",
		  "name": "Eleven",
		  "deviceType": "SMARTPHONE",
		  "manufacturer": "640",
		  "MAC": "E8:FB:E9:0C:2C:80",
		  "signalStrength": "0",
		  "rssi": "0",
		  "supportedProfile": "Not Identified;Not Identified;..."
		  },
		  "success": true
		  }
		}	

get Audio Info	curl http://127.0.0.1:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":"3", "method":"org.rdk.Bluetooth.1.getAudioInfo", "params": {"deviceID": "256168644324480"}}'	{
		  "jsonrpc": "2.0",
		  "id": 3,
		  "result": {
		  "trackInfo": {
		  "album": "Spacebound Apes",
		  "genre": "Jazz",
		  "title": "Grace",
		  "artist": "Neil Cowley Trio",
		  "ui32Duration": "217292",
		  "ui32TrackNumber": "1",
		  "ui32NumberOfTracks": "73"
		  },
		  "success": true
		  }
		}	

send Audio Playback Command	curl http://127.0.0.1:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":"3", "method":"org.rdk.Bluetooth.1.sendAudioPlaybackCommand", "params": {"deviceID": "256168644324480", "command": "PLAY"}}'	{"jsonrpc":"2.0","id":3,"result":{"success":true}}	
respond To Event	curl http://127.0.0.1:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0","id": "3", "method":"org.rdk.Bluetooth.1.respondToEvent", "params": {"deviceID": "256168644324480", "eventType": "onPairingRequest", "responseValue": "ACCEPTED"}}'	{"jsonrpc":"2.0","id":3,"result":{"success":true}}	

Note: The functions are referred from the RDK Service's Bluetooth wiki page

org.rdk.DisplaySettings

Prerequisite: This service has to be enabled from the controller UI to proceed with the execution of the below APIs

Function	Request	Response	Remarks

getConnectedVideoDisplays	curl http://127.0.0.1:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.DisplaySettings.1.getConnectedVideoDisplays"}'	{"jsonrpc":"2.0","id":3,"result":{"connectedVideoDisplays":["HDMI0"],"success":true}}	
getConnectedAudioPorts	curl http://127.0.0.1:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.DisplaySettings.1.getConnectedAudioPorts"}'	{"jsonrpc":"2.0","id":3,"result":{"connectedAudioPorts":["HDMI0"],"success":true}}	
getSupportedResolutions	curl http://127.0.0.1:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.DisplaySettings.1.getSupportedResolutions", "params":{"videoDisplay":"HDMI0"}}'	{"jsonrpc":"2.0","id":3,"result":{"supportedResolutions":["720p","1080i","1080p60"],"success":true}}	
getSupportedVideoDisplays	curl http://127.0.0.1:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.DisplaySettings.1.getSupportedVideoDisplays"}'	{"jsonrpc":"2.0","id":3,"result":{"supportedVideoDisplays":["HDMI0"],"success":true}}	
getSupportedTvResolutions	curl http://127.0.0.1:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.DisplaySettings.1.getSupportedTvResolutions", "params":{"videoDisplay":"HDMI0"}}'	{"jsonrpc":"2.0","id":3,"result":{"supportedTvResolutions":["480i","480p","576i","576p","720p","1080i","1080p"],"success":true}}	
getSupportedSettopResolutions	curl http://127.0.0.1:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.DisplaySettings.1.getSupportedSettopResolutions"}'	{"jsonrpc":"2.0","id":3,"result":{"supportedSettopResolutions":["480p","576p50","720p","720p50","1080i","1080p","1080i50", "1080p50","1080p24","1080p25","1080p30"],"success":true}}	
getSupportedAudioPorts	curl --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.DisplaySettings.1.getSupportedAudioPorts"}'	{"jsonrpc":"2.0","id":3,"result":{"supportedAudioPorts":["HDMI0"],"success":true}}	
getSupportedAudioModes	curl http://127.0.0.1:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.DisplaySettings.1.getSupportedAudioModes", "params":{"audioPort":"HDMI0"}}'	{"jsonrpc":"2.0","id":3,"result":{"supportedAudioModes":["STEREO","AUTO (Stereo)"],"success":true}}	
getZoomSetting	curl http://127.0.0.1:9998/jsonrpc --header "Content-Type: application/json" --request POST --data ' {"jsonrpc":"2.0", "id":3, "method":"org.rdk.DisplaySettings.1.getZoomSetting"}'	{"jsonrpc":"2.0","id":3,"result":{"zoomSetting":"FULL","success":true}}	
setZoomSetting	curl http://127.0.0.1:9998/jsonrpc --header "Content-Type: application/json" --request POST --data ' {"jsonrpc":"2.0", "id":3, "method":"org.rdk.DisplaySettings.1.setZoomSetting", "params":{"zoomSetting":"FULL"}}'	{"jsonrpc":"2.0", "id":3, "result":{"success":true}}	
getCurrentResolution	curl http://127.0.0.1:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.DisplaySettings.1.getCurrentResolution", "params":{"videoDisplay":"HDMI0"}}'	{"jsonrpc":"2.0","id":3,"result":{"resolution":"720p","success":true}}	
setCurrentResolution	curl http://127.0.0.1:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.DisplaySettings.1.setCurrentResolution", "params":{"videoDisplay":"HDMI0", "resolution":"720p", "persist":true}}'	{"jsonrpc":"2.0","id":3,"result":{"success":true}}	
getSoundMode	curl http://127.0.0.1:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.DisplaySettings.1.getSoundMode", "params":{"audioPort":"HDMI0"}}'	{"jsonrpc":"2.0","id":3,"result":{"soundMode":"STEREO","success":true}}	
setSoundMode	curl http://127.0.0.1:9998/jsonrpc --header "Content-Type: application/json" --request POST --data ' {"jsonrpc":"2.0", "id":3, "method":"org.rdk.DisplaySettings.1.setSoundMode", "params":{"audioPort":"HDMI0", "soundMode":"stereo", "persist":true}}'	{"jsonrpc":"2.0","id":3,"result":{"success":true}}	
readEDID	curl http://127.0.0.1:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.DisplaySettings.1.readEDID"}'	{"jsonrpc":"2.0","id":3,"result":{"EDID":"AP ///////wAQrMLQVEJTMQUdAQOA...","success":true}}	
readHostEDID	curl http://127.0.0.1:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.DisplaySettings.1.readHostEDID"}'	{"jsonrpc":"2.0","id":3,"result":{"EDID":"AP ///////wAIbSV0Z0UjARgYAQOAgEh...","success":true}}	
getActiveInput	curl http://127.0.0.1:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.DisplaySettings.1.getActiveInput", "params":{"videoDisplay":"HDMI0"}}'	{"jsonrpc":"2.0","id":3,"result":{"activeInput":false,"success":true}}	
getTvHDRSupport	curl http://127.0.0.1:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.DisplaySettings.1.getTvHDRSupport"}'	{"jsonrpc":"2.0","id":3,"result":{"supportsHDR":false,"standards":["none"],"success":true}}	
getSettopHDRSupport	curl http://127.0.0.1:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.DisplaySettings.1.getSupportedSettopResolutions"}'	{"jsonrpc":"2.0","id":3,"result":{"supportedSettopResolutions":["720p","1080i","1080p60","2160p30","2160p60"],"success":true}}	
setVideoPortStatusInStandby	curl http://127.0.0.1:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.DisplaySettings.1.setVideoPortStatusInStandby", "params":{"portName":"HDMI0", "enabled":true}}'	{"jsonrpc":"2.0","id":3,"result":{"success":true}}	
getVideoPortStatusInStandby	curl http://127.0.0.1:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.DisplaySettings.1.getVideoPortStatusInStandby", "params":{"portName":"HDMI0"}}'	{"jsonrpc":"2.0","id":3,"result":{"videoPortStatusInStandby":false,"success":true}}	
getCurrentOutputSettings	curl http://127.0.0.1:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.DisplaySettings.1.getCurrentOutputSettings"}'	{"jsonrpc":"2.0","id":3,"result":{"colorSpace":5,"colorDepth":0,"matrixCoefficients":0,"videoEOTF":0,"success":true}}	

setVolumeLeveller	curl http://127.0.0.1:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.DisplaySettings.1.setVolumeLeveller", "params":{"audioPort":"SPEAKER0","level":"9"}}'	{"jsonrpc":"2.0","id":3,"result":{"success":true}}	
getVolumeLeveller	curl http://127.0.0.1:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.DisplaySettings.1.getVolumeLeveller", "params":{"audioPort":"SPEAKER0"}'	{"jsonrpc":"2.0","id":3,"result":{"enable":true,"level":9,"success":true}}	
setBassEnhancer	curl http://127.0.0.1:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.DisplaySettings.1.setBassEnhancer", "params":{"audioPort":"SPEAKER0","bassEnhancerEnable":true}}'	{"jsonrpc":"2.0","id":3,"result":{"success":true}}	
getBassEnhancer	curl http://127.0.0.1:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.DisplaySettings.1.getBassEnhancer", "params":{"audioPort":"SPEAKER0"}'	{"jsonrpc":"2.0","id":3,"result":{"bassEnhancerEnable":true,"success":true}}	
setSurroundVirtualizer	curl http://127.0.0.1:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.DisplaySettings.1.setSurroundVirtualizer", "params":{"audioPort":"SPEAKER0","boost":"90"}}'	{"jsonrpc":"2.0","id":3,"result":{"success":true}}	
getSurroundVirtualizer	curl http://127.0.0.1:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.DisplaySettings.1.getSurroundVirtualizer", "params":{"audioPort":"SPEAKER0"}'	{"jsonrpc":"2.0","id":3,"result":{"enable":true,"boost":90,"success":true}}	
enableSurroundDecoder	curl http://127.0.0.1:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.DisplaySettings.1.enableSurroundDecoder", "params":{"audioPort":"SPEAKER0","surroundDecoderEnable":true}}'	{"jsonrpc":"2.0","id":3,"result":{"success":true}}	
isSurroundDecoderEnabled	curl http://127.0.0.1:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.DisplaySettings.1.isSurroundDecoderEnabled", "params":{"audioPort":"SPEAKER0"}'	{"jsonrpc":"2.0","id":3,"result":{"surroundDecoderEnable":true,"success":true}}	
setDRCMode	curl http://127.0.0.1:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.DisplaySettings.1.setDRCMode", "params":{"audioPort":"SPEAKER0","DRCMode":"1"}}'	{"jsonrpc":"2.0","id":3,"result":{"success":true}}	
getDRCMode	curl http://127.0.0.1:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.DisplaySettings.1.getDRCMode", "params":{"audioPort":"SPEAKER0"}'	{"jsonrpc":"2.0","id":3,"result":{"DRCMode":"line","success":true}}	
setVolumeLevel	curl http://127.0.0.1:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.DisplaySettings.1.setVolumeLevel", "params":{"audioPort":"HDMI0","volumeLevel":"20"}}'	{"jsonrpc":"2.0","id":3,"result":{"success":true}}	
getVolumeLevel	curl http://127.0.0.1:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.DisplaySettings.1.getVolumeLevel", "params":{"audioPort":"SPEAKER0"}'	{"jsonrpc":"2.0","id":3,"result":{"enable":true,"volumeLevel":50,"success":true}}	
setMuted	curl http://127.0.0.1:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.DisplaySettings.1.setMuted", "params":{"audioPort":"SPEAKER0","muted":true}}'	{"jsonrpc":"2.0","id":3,"result":{"success":true}}	
getMuted	curl http://127.0.0.1:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.DisplaySettings.1.getMuted", "params":{"audioPort":"SPEAKER0"}'	{"jsonrpc":"2.0","id":3,"result":{"muted":true,"success":true}}	
getAudioDelay	curl http://127.0.0.1:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.DisplaySettings.1.getAudioDelay", "params":{"audioPort":"HDMI0"}}'	{"jsonrpc":"2.0","id":3,"result":{"audioDelay":"0","success":true}}	
setAudioDelay	curl http://127.0.0.1:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.DisplaySettings.1.setAudioDelay", "params":{"audioDelay":"50", "audioPort":"HDMI0"}}'	{"jsonrpc":"2.0","id":3,"result":{"success":true}}	
getAudioDelayOffset	curl http://127.0.0.1:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.DisplaySettings.1.getAudioDelayOffset", "params":{"audioPort":"HDMI0"}}'	{"jsonrpc":"2.0","id":3,"result":{"audioDelayOffset":"0","success":true}}	
setAudioDelayOffset	curl http://127.0.0.1:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.DisplaySettings.1.setAudioDelayOffset", "params":{"audioDelayOffset":"50", "audioPort":"HDMI0"}}'	{"jsonrpc":"2.0","id":3,"result":{"success":true}}	
getSinkAtmosCapability	curl http://127.0.0.1:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.DisplaySettings.1.getSinkAtmosCapability"}'	{"jsonrpc":"2.0","id":3,"result":{"atmos_capability":2,"success":true}}	
setAudioAtmosOutputMode	curl http://127.0.0.1:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.DisplaySettings.1.setAudioAtmosOutputMode", "params":{"enable":true}}'	{"jsonrpc":"2.0","id":3,"result":{"success":true}}	
getTVHDRCapabilities	curl http://127.0.0.1:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.DisplaySettings.1.getTVHDRCapabilities"}'	{"jsonrpc":"2.0","id":3,"result":{"capabilities":3,"success":true}}	
getDefaultResolution	curl http://127.0.0.1:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.DisplaySettings.1.getDefaultResolution"}'	{"jsonrpc":"2.0","id":3,"result":{"defaultResolution":"720p","success":true}}	
setScartParameter	curl http://127.0.0.1:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.DisplaySettings.1.setScartParameter", "params":{"scartParameter":"aspect_ratio", "scartParameterData":"4x3"}}'	{"jsonrpc":"2.0","id":3,"result":{"success":false}}	

Note: The functions are referred from the RDK Service's Display Settings wiki page

org.rdk.System

Prerequisite: This service has to be enabled from the controller UI to proceed with the execution of the below APIs

Function	Request	Response	Remarks
cacheContains	curl http://localhost:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0","id":"3","method":"org.rdk.System.1.cacheContains","params":{"key":"sampleKey"}}'	{"jsonrpc":"2.0","id":3,"result":{"success":true}}	
clearLastDeepSleepReason	curl http://localhost:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0","id":"3","method":"org.rdk.System.1.clearLastDeepSleepReason","params":{}}'	{"jsonrpc":"2.0","id":3,"result":{"success":true}}	
getAvailableStandbyModes	curl http://localhost:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0","id":"3","method":"org.rdk.System.1.getAvailableStandbyModes","params":{}}'	{"jsonrpc":"2.0","id":3,"result":{"supportedStandbyModes":["LIGHT_SLEEP","DEEP_SLEEP"],"success":true}}	
getCachedValue	curl http://localhost:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0","id":"3","method":"org.rdk.System.1.getCachedValue","params":{"key":"sampleKey"}}'	{"jsonrpc":"2.0","id":3,"result":{"sampleKey":"4343.3434","success":true}}	
getCoreTemperature	curl http://localhost:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0","id":"3","method":"org.rdk.System.1.getCoreTemperature","params":{}}'	{"jsonrpc":"2.0","id":3,"result":{"temperature":48.000000,"success":true}}	
getDeviceInfo	curl http://localhost:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0","id":"3","method":"org.rdk.System.1.getDeviceInfo","params":{"params":["estb_mac"]}}'	{"jsonrpc":"2.0","id":3,"result":{"ecm_mac/estb_mac":"AA:BB:CC:DD:EE:FF:GG","success":true}}	
getDownloadedFirmwareInfo	curl http://localhost:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0","id":"3","method":"org.rdk.System.1.getDownloadedFirmwareInfo","params":{}}'	{"jsonrpc":"2.0","id"3,"result" {"currentFWVersion":"AX061AEI_VBN_1911_sprint_20200109040424sdy", "downloadedFWVersion":"AX061AEI_VBN_1911_sprint_20200510040450sdy", "downloadedFWLocation":"","isRebootDeferred":false,"success":true}}	
getFirmwareDownloadPercent	curl http://localhost:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0","id":"3","method":"org.rdk.System.1.getFirmwareDownloadPercent","params":{}}'	{"jsonrpc":"2.0","id":3,"result":{"downloadPercent":25,"success":true}}	
getFirmwareUpdateInfo	curl http://localhost:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0","id":"3","method":"org.rdk.System.1.getFirmwareUpdateInfo","params":{"param":{"GUID":"1234abcd"}}}'	{"jsonrpc":"2.0","id":3,"result":{"asyncResponse":true}}	
getFirmwareUpdateState	curl http://localhost:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0","id":"3","method":"org.rdk.System.1.getFirmwareUpdateState","params":{}}'	{"jsonrpc":"2.0","id":3,"result":{"firmwareUpdateState":"Download Complete","success":true}}	
getLastDeepSleepReason	curl http://localhost:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0","id":"3","method":"org.rdk.System.1.getLastDeepSleepReason","params":{}}'	{"jsonrpc":"2.0","id":3,"result":{"lastDeepSleepReason":"thermal_deepsleep_critical_threshold","success":true}}	
getMacAddresses	curl http://localhost:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0","id":"3","method":"org.rdk.System.1.getMacAddresses","params":{"GUID":"61734787891723481"}}'	{"jsonrpc":"2.0","id":3,"result":{"asyncResponse":true,"success":true}}	

getMilestones	curl http://localhost:9998/jsonrpc --header "Content-Type: application /json" --request POST --data '{"jsonrpc":"2.0","id":"3","method": "org.rdk.System.1.getMilestones","params":{}}'	{"jsonrpc":"2.0","id":3,"result":{"milestones":["RDK_STARTED:8893","SEND_DHCP_REQUEST:9530","RECV_DHCP_RESPONSE:9602","UPNP_START_DISCOVERY:12863","UPNP_START_DISCOVERY:13618","IP_ACQUISTION_COMPLETED:15716","IP_ACQUISTION_COMPLETED:16470","2020 Sep 24 20:32:49.688454 raspberrypi-rdk-mc systemd[1]: Starting Log RDK Started Service...","-- Subject: Unit log-rdk-start.service has begun start-up","-- Defined-By: systemd","-- Support: http://lists.freedesktop.org/mailman/listinfo/systemd-devel","-- ","-- Unit log-rdk-start.service has begun starting up.","2020 Sep 24 20:32:50.088253 raspberrypi-rdk-mc systemd[1]: Started Log RDK Started Service.","-- Subject: Unit log-rdk-start.service has finished start-up","-- Defined-By: systemd","-- Support: http://lists.freedesktop.org/mailman/listinfo/systemd-devel","-- ","-- Unit log-rdk-start.service has finished starting up.","-- ","-- The start-up result is done."],"success":true}}	
getMode	curl http://localhost:9998/jsonrpc --header "Content-Type: application /json" --request POST --data '{"jsonrpc":"2.0","id":"3","method":" org.rdk.System.1.getMode","params":{}}'	{"jsonrpc":"2.0","id":3,"result":{"modeInfo":{"mode":"NORMAL","duration":0},"success":true}}	
getPowerState	curl http://localhost:9998/jsonrpc --header "Content-Type: application /json" --request POST --data '{"jsonrpc":"2.0","id":"3","method":" org.rdk.System.1.getPowerState","params":{}}'	{"jsonrpc":"2.0","id":3,"result":{"powerState":"ON", "success":true}}	
getPreferredStandbyMode	curl http://localhost:9998/jsonrpc --header "Content-Type: application /json" --request POST --data '{"jsonrpc":"2.0","id":"3","method":" org.rdk.System.1. getPreferredStandbyMode","params":{}}'	{"jsonrpc":"2.0","id":3,"result":{"preferredStandbyMode":"DEEP_SLEEP","success":true}}	
getPreviousRebootInfo	curl http://localhost:9998/jsonrpc --header "Content-Type: application /json" --request POST --data '{"jsonrpc":"2.0","id":"3","method":" org.rdk.System.1. getPreviousRebootInfo","params":{}}'	{"jsonrpc":"2.0","id":3,"result":{"timeStamp":"20200128083540","reason":"Triggered from System Plugin API Validation","success":true}}	
getPreviousRebootInfo2	curl http://localhost:9998/jsonrpc --header "Content-Type: application /json" --request POST --data '{"jsonrpc":"2.0","id":"3","method":" org.rdk.System.1. getPreviousRebootInfo2","params":{}}'	{"jsonrpc":"2.0","id":3,"result":{"rebootInfo":{"timestamp":"Tue Jan 28 08:35:40 UTC 2020","source":"SystemPlugin","reason":"FIRMWARE_FAILURE","customReason":"API Validation","lastHardPowerReset":"Tue Jan 28 08:22:22 UTC 2020"},"success":true}}	
getPreviousRebootReason	curl http://localhost:9998/jsonrpc --header "Content-Type: application /json" --request POST --data '{"jsonrpc":"2.0","id":"3","method":" org.rdk.System.1. getPreviousRebootReason","params":{}}'	{"jsonrpc":"2.0","id":3,"result":{"RFCConfig":{"Device.DeviceInfo.X_RDKCENTRAL-COM_RFC.Feature.AccountInfo.AccountID":"11814056142354993 71","Device.DeviceInfo.X_RDKCENTRAL-COM_RFC.Feature.UPnP.Enabled":"Empty response received","Device.DeviceInfo.X_RDKCENTRAL-COM_RFC.Feature.AllowOpenPorts.Enabled":"Empty response received","Device.DeviceInfo.MemoryStatus.Total":"1144560","Device.DeviceInfo.MemoryStatus.Free":"732144"},"success":true}}	
getRFCConfig	curl http://localhost:9998/jsonrpc --header "Content-Type: application /json" --request POST --data '{"jsonrpc":"2.0","id":"3","method":" org.rdk.System.1.getRFCConfig"," params":{"rfcList":["Device.DeviceInfo. X_RDKCENTRAL-COM_RFC.Feature. AccountInfo.AccountID","Device. DeviceInfo.X_RDKCENTRAL-COM_RFC. Feature.UPnP.Enabled","Device. DeviceInfo.X_RDKCENTRAL-COM_RFC. Feature.AllowOpenPorts.Enabled"," Device.DeviceInfo.MemoryStatus.Total"," Device.DeviceInfo.MemoryStatus. Free"]}}'	{"jsonrpc":"2.0","id":3,"result":{"RFCConfig":{"Device.DeviceInfo.X_RDKCENTRAL-COM_RFC.Feature.AccountInfo.AccountID":"11814056142354993 71","Device.DeviceInfo.X_RDKCENTRAL-COM_RFC.Feature.UPnP.Enabled":"Empty response received","Device.DeviceInfo.X_RDKCENTRAL-COM_RFC.Feature.AllowOpenPorts.Enabled":"Empty response received","Device.DeviceInfo.MemoryStatus.Total":"1144560","Device.DeviceInfo.MemoryStatus.Free":"732144"},"success":true}}	
getSerialNumber	curl http://localhost:9998/jsonrpc --header "Content-Type: application /json" --request POST --data '{"jsonrpc":"2.0","id":"3","method":" org.rdk.System.1.getSerialNumber"," params":{}}'	{"jsonrpc":"2.0","id":3,"result":{"serialNumber":"M11806TK0024","success": true}}	
getStateInfo	curl http://localhost:9998/jsonrpc --header "Content-Type: application /json" --request POST --data '{"jsonrpc":"2.0","id":"3","method":" org.rdk.System.1.getStateInfo"," params":{"param":"com.comcast. channel_map"}}'	{"jsonrpc":"2.0","id":3,"result":{"com.comcast.channel_map”:2, “success”: true}}	
getSystemVersions	curl http://localhost:9998/jsonrpc --header "Content-Type: application /json" --request POST --data '{"jsonrpc":"2.0","id":"3","method":" org.rdk.System.1.getSystemVersions"," params":{}}'	{"jsonrpc":"2.0","id":3,"result":{"stbVersion":" AX061AEI_VBN_1911_sprint_20200109040424sdy","receiverVersion":"3.14.0.0"," stbTimestamp":"Thu 09 Jan 2020 04:04:24 AP UTC","success":true}}	
getTemperatureThresholds	curl http://localhost:9998/jsonrpc --header "Content-Type: application /json" --request POST --data '{"jsonrpc":"2.0","id":"3","method":" org.rdk.System.1. getTemperatureThresholds","params":{}}'	{"jsonrpc":"2.0","id":3,"result":{"temperatureThresholds":{"WARN":" 100.000000","MAX":"110.000000","temperature":"48.000000"},"success":true}}	

getTimeZoneDST	curl http://localhost:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0","id":"3","method":"org.rdk.System.1.getTimeZoneDST","params":{}}'	{"jsonrpc":"2.0","id":3,"result":{"timeZone":"America/New_York","success":true}}	
getXconfParams	curl http://localhost:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0","id":"3","method":"org.rdk.System.1.getXconfParams","params":{}}'	{"jsonrpc":"2.0","id":3,"result":{"xconfParams":{"env":"vbn","eStbMac":"20:F1:9E:EE:62:08","model":"AX061AEI","firmwareVersion":"AX061AEI_VBN_1911_sprint_20200109040424sdy"},"success":true}}	
hasRebootBeenRequested	curl http://localhost:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0","id":"3","method":"org.rdk.System.1.hasRebootBeenRequested","params":{}}'	{"jsonrpc":"2.0","id":3,"result":{"rebootRequested":false,"success":true}}	
isGzEnabled	curl http://localhost:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0","id":"3","method":"org.rdk.System.1.isGzEnabled","params":{}}'	{"jsonrpc":"2.0","id":3,"result":{"enabled":false,"success":true}}	
queryMocaStatus	curl http://localhost:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0","id":"3","method": "org.rdk.System.1.queryMocaStatus","params":{}}'	{"jsonrpc":"2.0","id":3,"result":{"mocaEnabled":true,"success":true}}	
reboot	curl http://localhost:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0","id":"3","method":"org.rdk.System.1.reboot","params":{"rebootReason":"API Validation"}}'	{"jsonrpc":"2.0","id":3,"result":{"IARM_Bus_Call_STATUS":1,"success":true}}	
removeCacheKey	curl http://localhost:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0","id":"3","method":"org.rdk.System.1.removeCacheKey","params":{key:"sampleKey"}}'	{"jsonrpc":"2.0","id":3,"result":{"success":true}}	
requestSystemUptime	curl http://localhost:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0","id":"3","method":"org.rdk.System.1.requestSystemUptime","params":{}}'	{"jsonrpc":"2.0","id":3,"result":{"systemUptime":"1666.92","success":true}}	
setCachedValue	curl http://localhost:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0","id":"3","method":"org.rdk.System.1.setCachedValue","params":{"key":"sampleKey","value":4343.3434}}'	{"jsonrpc":"2.0","id":3,"result":{"success":true}}	
setDeepSleepTimer	curl http://localhost:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0","id":"3","method":"org.rdk.System.1.setDeepSleepTimer","params":{"seconds":3}}'	{"jsonrpc":"2.0","id":3,"result":{"success":true}}	
setGzEnabled	curl http://localhost:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0","id":"3","method":"org.rdk.System.1.setGzEnabled","params":{"enabled":true}}'	{"jsonrpc":"2.0","id":3,"result":{"success":true}}	
set STB mode	curl http://localhost:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0","id":"3","method":"org.rdk.System.1.setGzEnabled","params":{"modeInfo":{"mode":"NORMAL","duration":20}}}'	{"jsonrpc":"2.0","id":3,"result":{"success":true}}	
setGzEnabled	curl http://localhost:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0","id":"3","method":"org.rdk.System.1.setPowerState","params":{"powerState":"STANDBY", "standbyReason":"APIUnitTest"}}'	{"jsonrpc":"2.0","id":3,"result":{"success":true}}	
setPreferredStandbyMode	curl http://localhost:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0","id":"3","method":"org.rdk.System.1.setPreferredStandbyMode","params":{"standbyMode":"DEEP_SLEEP"}}'	{"jsonrpc":"2.0","id":3,"result":{"success":true}}	
setTemperatureThresholds	curl http://localhost:9998/jsonrpc --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0","id":"3","method":"org.rdk.System.1.setTemperatureThresholds","params":{"thresholds":{"WARN":"50.000000","MAX":"80.000000"}}}'	{"jsonrpc":"2.0","id":3,"result":{"success":true}}	

setTimeZone DST	curl http://localhost:9998/jsonrpc --header "Content-Type: application /json" --request POST --data '{"jsonrpc" :"2.0","id":"3","method":"org.rdk. System.1.setTimeZoneDST","params": {"timeZone":"UTC-5"}}'	{"jsonrpc":"2.0","id":3,"result":{"success":true}}	
updateFirmw are	curl http://localhost:9998/jsonrpc --header "Content-Type: application /json" --request POST --data '{"jsonrpc" :"2.0","id":"3","method":"org.rdk. System.1.updateFirmware","params":{}}'	{"jsonrpc":"2.0","id":3,"result":{"success":true}}	

Note: The functions are referred from the RDK Service's System wiki page

FirmwareControl

Prerequisite: This service has to be enabled from the controller UI to proceed with the execution of the below APIs

Function	Request	Response	Remarks
upgrade	curl --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.FirmwareControl.1.upgrade", "params":{"name": "firmware_v.0", "location": "http://my .site.com/images", "type": "RCDL", "progressinterval": 10, "hmac": "2834e6d07fa4c7778ef7a4e394f38a5c321afbed51d54ad512bd3fffbc7aa5debc"}}' http://127.0.0.1:9998/jsonrpc	{ "jsonrpc": " 2.0", "id": 12 34567890, "res ult": null }	
resume	curl --header "Content-Type: application/json" --request POST --data ' { "jsonrpc": "2.0", "id": 1234567890, "method": "FirmwareControl.1.resume", "params": { "name": "firmware_v.0", "location": "http://my.site.com/images" } } ' http://127.0.0.1:9998/jsonrpc	{ "jsonrpc": "2.0", "id" : 1234567890, "resu lt": null }	

Note: The functions are referred from the Thunder Nano Service's GitHub link

org.rdk.FrameRate

Function	Request	Response	Remarks
set FPS log frequency	curl --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0","id":"3" ,"params": {"frequencyInMs":1000},"method": "FrameRate.1.setCollectionFrequency"}' http://127. 0.0.1:9998/jsonrpc	{"jsonrpc":"2.0","id":3 ,"result":{"success":t rue}}	
set default FPS log frequency	curl --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0","id":"3" ,"method": "FrameRate.1.setCollectionFrequency"}' http://127.0.0.1:9998/jsonrpc		
start FPS Collection	curl --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0","id":"3" ,"method": "FrameRate.1.startFpsCollection"}' http://127.0.0.1:9998/jsonrpc		
stop FPS Collection	curl --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0","id":"3" ,"method": "FrameRate.1.stopFpsCollection"}' http://127.0.0.1:9998/jsonrpc		

Note: The functions are referred from the RDK Service's Frame Rate wiki page

org.rdk.HDCPProfile

Prerequisite: This service has to be enabled from the controller UI to proceed with the execution of the below APIs

Function	Request	Response	Remarks
get HDCP status	curl --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.HdcpProfile.1. getHDCPStatus"}' http://127.0.0.1:9998/jsonrpc	{"jsonrpc":"2.0","id":3,"result":{"HDCPStat us":{"isConnected":false,"isHDCPCompliant": false,"isHDCPEnabled":false, "supportedHDCPVersion":"2.2","receiverHDCPV ersion":"1.4","currentHDCPVersion":"1.4"}, "success":true}}	RPI doesn't have any input device for basic testing
get set Top HDCP Support	curl --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.HdcpProfile.1. getSettopHDCPSupport"}' http://127.0.0.1:9998/jsonrpc	{"jsonrpc":"2.0","id":3,"result":{"supporte dHDCPVersion":"2.2","isHDCPSupported":true, "success":true}}	

Note: The functions mentioned above are referred from the RDK Service's HDCPProfile wiki page

org.rdk.HDMIInput

Prerequisite: This service has to be enabled from the controller UI to proceed with the execution of the below APIs

Function	Request	Response	Remarks
get HDMI Input Devices	curl --header "Content-Type: application/json" --request POST -- data '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.HdmiInput.1. getHDMIInputDevices"}' http://127.0.0.1:9998/jsonrpc	{"jsonrpc":"2.0","id":3,"result":{"devices": [{"id":0,"locator":"hdmiin://localhost /deviceid/0","connected":"false"}],"success": true}}	

start HDMI Input	curl --header "Content-Type: application/json" --request POST --data '{" jsonrpc":"2.0", "id":3, "method":"org.rdk.HdmiInput.1.startHdmiInput", "params": {"portId":"1"}}' http://127.0.0.1:9998/jsonrpc	{"jsonrpc":"2.0","id":3,"result":{"success": true}}	
stop HDMI Input	curl --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.HdmiInput.1. stopHdmiInput"}' http://127.0.0.1:9998/jsonrpc	{"jsonrpc":"2.0","id":3,"result":{"success": true}}	
write EDID	curl --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.HdmiInput.1.writeEDID", "params": {"deviceId": 0, "message": "EDID"}}' http://127.0.0.1:9998 /jsonrpc	{"jsonrpc":"2.0","id":3,"result":{"name":" HdmiInputEDIDStub","success":true}}	
read EDID	curl --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.HdmiInput.1.readEDID"}' htt p://127.0.0.1:9998/jsonrpc	{"jsonrpc":"2.0","id":3,"result":{"name":" HdmiInputEDIDStub","success":true}}	

Note: The functions mentioned above are referred from the RDK Service's HDMI Input wiki page

org.rdk.LoggingPreferences

Prerequisite: This service has to be enabled from the controller UI to proceed with the execution of the below APIs

Function	Request	Response	Remarks
is Key Stroke Mask Enabled	curl --header "Content-Type: application/json" --request POST --data '{"jsonrpc":" 2.0", "id":3, "method":"org.rdk.LoggingPreferences.1.isKeystrokeMaskEnabled"}' http:/ /127.0.0.1:9998/jsonrpc	{"jsonrpc":"2.0","id":3," result": {"keystrokeMaskEnabled": true,"success":true}}	RPI doesn't have any input device for basic testing
set Key Stroke Mask Enabled	curl --header "Content-Type: application/json" --request POST --data '{"jsonrpc":" 2.0", "id":3, "method":"org.rdk.LoggingPreferences.1.setKeystrokeMaskEnabled", "params":{"keystrokeMaskEnabled":true}}' http://127.0.0.1:9998/jsonrpc	{"jsonrpc":"2.0","id":3," result":{"success":true}}	

Note: The functions mentioned above are referred from the RDK Service's Logging Preferences wiki page

org.rdk.AVInput

Prerequisite: This service has to be enabled from the controller UI to proceed with the execution of the below APIs

Function	Request	Response	Remarks
get Number of Input devices	curl --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0","id":"3","method": "org.rdk.AVInput.1.numberOfInputs"}' http://127.0.0.1:9998/jsonrpc	{"jsonrpc":"2.0","id":3,"result": {"numberOfInputs":0,"message":" Success","success":false}}	RPI doesn't have any input device for basic testing
current Video mode	curl --header "Content-Type: application/json" --request POST --data '{"js onrpc":"2.0","id":"3","method": "org.rdk.AVInput.1.currentVideoMode"}' htt p://127.0.0.1:9998/jsonrpc	{"jsonrpc":"2.0","id":3,"result":{"cu rrentVideoMode":"unknownp","success": true}}	
is Content Protected	curl --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0","id":"3","method": "org.rdk.AVInput.1. contentProtected"}' http://127.0.0.1:9998/jsonrpc	{"jsonrpc":"2.0","id":3,"result": {"isContentProtected":true,"success": true}}}	

Note: The functions mentioned above are referred from the RDK Service's AV Input wiki page

org.rdk.ActivityMonitor

Prerequisite: This service has to be enabled from the controller UI to proceed with the execution of the below APIs

Function	Request	Response	Remarks
enable Monitoring	curl --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.ActivityMonitor.1. enableMonitoring", "params":{"config":[{"appPid":"6860", "memoryThresholdMB":"1", "cpuThresholdPercent":"1", "cpuThresholdSeconds":"1"}], "memoryIntervalSeconds":"0.02", "cpuIntervalSeconds":"0.02"}}' http://127.0.0.1:9998/jsonrpc	{"jsonrpc":"2.0", "id":3, "result": {"success":true}}	correct pid to be given
disable Monitoring	curl --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.ActivityMonitor.1. disableMonitoring", "params":{}}' http://127.0.0.1:9998/jsonrpc	{"jsonrpc":"2.0", "id":3, "result": {"success":true}}	
get Application Memory Usage	curl --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.ActivityMonitor.1. getApplicationMemoryUsage", "params":{"pid":6860}}' http://127.0.0.1:9998 /jsonrpc	{"jsonrpc":"2.0", "id":3, "result": {"applicationMemory":{"appPid":6860," appName":"WPEProcess","memoryMB":6}," success":true}}	correct Pid to be given
get All Apps Memory Usage	curl --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.ActivityMonitor.1. getAllMemoryUsage", "params":{}}' http://127.0.0.1:9998/jsonrpc	{"jsonrpc":"2.0", "id":3, "result": {"free MemoryMB":795,"applicationMemory":[{"appPi d":7897,"appName":"WPENetworkProce","memor yMB":5},{"appPid":7899,"appName":"WPEWebPr ocess","memoryMB":19}],"success":true}}	

Note: The functions are referred from the RDK Service's ActivityMonitor wiki page

org.rdk.ScreenCapture

Prerequisite: This service has to be enabled from the controller UI to proceed with the execution of the below APIs

Function	Request	Response	Remarks
upload capture screen	curl --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":3, "method":"com.comcast.ScreenCapture.1.uploadScreenCapture", "params": {"url": "http://server/cgi-bin/upload.cgi"}}' http://127.0.0.1:9998/jsonrpc	{"jsonrpc":"2.0","id":3,"result":{"success":true}}	

Note: The functions are referred from the RDK Service's Screen Capture wiki page

org.rdk.Timer

Prerequisite: This service has to be enabled from the controller UI to proceed with the execution of the below APIs

Function	Request	Response	Remarks
start Timer	curl --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.Timer.1.startTimer", "params":{"interval":1.2, "mode":"WAKE", "repeatInterval":1.4, "remindBefore":0.2}}' http://127.0.0.1:9998/jsonrpc	{"jsonrpc":"2.0","id":3,"result":{"timerId":0,"success":true}}	
cancel	curl --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.Timer.1.cancel", "params":{"timerId":0}}' http://127.0.0.1:9998/jsonrpc	{"jsonrpc":"2.0","id":3,"result":{"success":true}}	
suspend	curl --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.Timer.1.suspend", "params":{"timerId":0}}' http://127.0.0.1:9998/jsonrpc	{"jsonrpc":"2.0","id":3,"result":{"success":true}}	
resume	curl --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.Timer.1.resume", "params":{"timerId":0}}' http://127.0.0.1:9998/jsonrpc	{"jsonrpc":"2.0","id":3,"result":{"success":true}}	
get Timer Status	curl --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.Timer.1.getTimerStatus", "params":{"timerId":0}}'http://127.0.0.1:9998/jsonrpc	{ "jsonrpc": "2.0", "id": 3, "result": { "state": "RUNNING", "mode": "WAKE", "timeRemaining": "1894.646", "repeatInterval": "2000.000", "remindBefore": "1000.000", "success": true } }	
get All Timers	curl --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.Timer.1.getTimers"}' http://127.0.0.1:9998/jsonrpc	{ "jsonrpc": "2.0", "id": 3, "result": { "timers": [{ "timerId": 0, "state": "RUNNING", "mode": "WAKE", "timeRemaining": "1984.411", "repeatInterval": "2000.000", "remindBefore": "1000.000" }], "success": true } }	

Note: The functions are referred from the RDK Service's Sleep/Wake Timer wiki page

org.rdk.UserPreferences

Prerequisite: This service has to be enabled from the controller UI to proceed with the execution of the below APIs

Function	Request	Response	Remarks
get UI Language	curl --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.UserPreferences.1.getUILanguage"}' http://127.0.0.1:9998/jsonrpc	{"jsonrpc":"2.0","id":3,"result":{"ui_language":"US_en","success":true}}	
set UI Language	curl --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.UserPreferences.1.setUILanguage", "params": {"ui_language": "US_en"}}' http://127.0.0.1:9998/jsonrpc	{"jsonrpc":"2.0","id":3,"result":{"success":true}}	

Note: The functions are referred from the RDK Service's User Preferences wiki page

org.rdk.WareHouse

Prerequisite: This service has to be enabled from the controller UI to proceed with the execution of the below APIs

Function	Request	Response	Remarks
resetDevice	curl --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.Warehouse.1.resetDevice", "params":{"suppressReboot":true}}' http://127.0.0.1:9998/jsonrpc	{"jsonrpc":"2.0","id":3,"result": {"PARAM_SUCCESS":true,"success":true}}	
getDeviceInfo	curl --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.Warehouse.1.getDeviceInfo"}' http://127.0.0.1:9998/jsonrpc	{"jsonrpc":"2.0","id":3,"result": {"bluetooth_mac":"00:00:00:00:00:00","boxIP":"192.168.0.145","build_type":"PI", "estb_mac":"B8:27:EB:C5:97:AF","eth_mac":"","imageVersion":"","version":"","software_version":"", "model_number":"pi","rf4ce_mac":"","wifi_mac":"","success":true}}	
internalReset	curl --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.Warehouse.1.internalReset", "params":{"passPhrase":"FOR TEST PURPOSES ONLY2"}}' http://127.0.0.1:9998/jsonrpc	{"jsonrpc":"2.0","id":3,"result": {"success":false,"error":"incorrect pass phrase"}}	
lightReset	curl --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.Warehouse.1.lightReset"}' http://127.0.0.1:9998/jsonrpc	{"jsonrpc":"2.0","id":3,"result": {"success":true}}	
isClean	curl --header "Content-Type: application/json" --request POST --data '{"jsonrpc":"2.0", "id":3, "method":"org.rdk.Warehouse.1.isClean"}' http://127.0.0.1:9998/jsonrpc	{"jsonrpc":"2.0","id":3,"result": {"success":true,"files":[],"clean": true}}	

Note: The functions are referred from the RDK Service's Warehouse wiki page