

REST API examples using cURL

© 2023 Imagine Communications Corp. Proprietary and Confidential.

Imagine Communications considers this document and its contents to be proprietary and confidential. Except for making a reasonable number of copies for your own internal use, you may not reproduce this publication, or any part thereof, in any form, by any method, for any purpose, or in any language other than English without the written consent of Imagine Communications. All other uses are illegal.

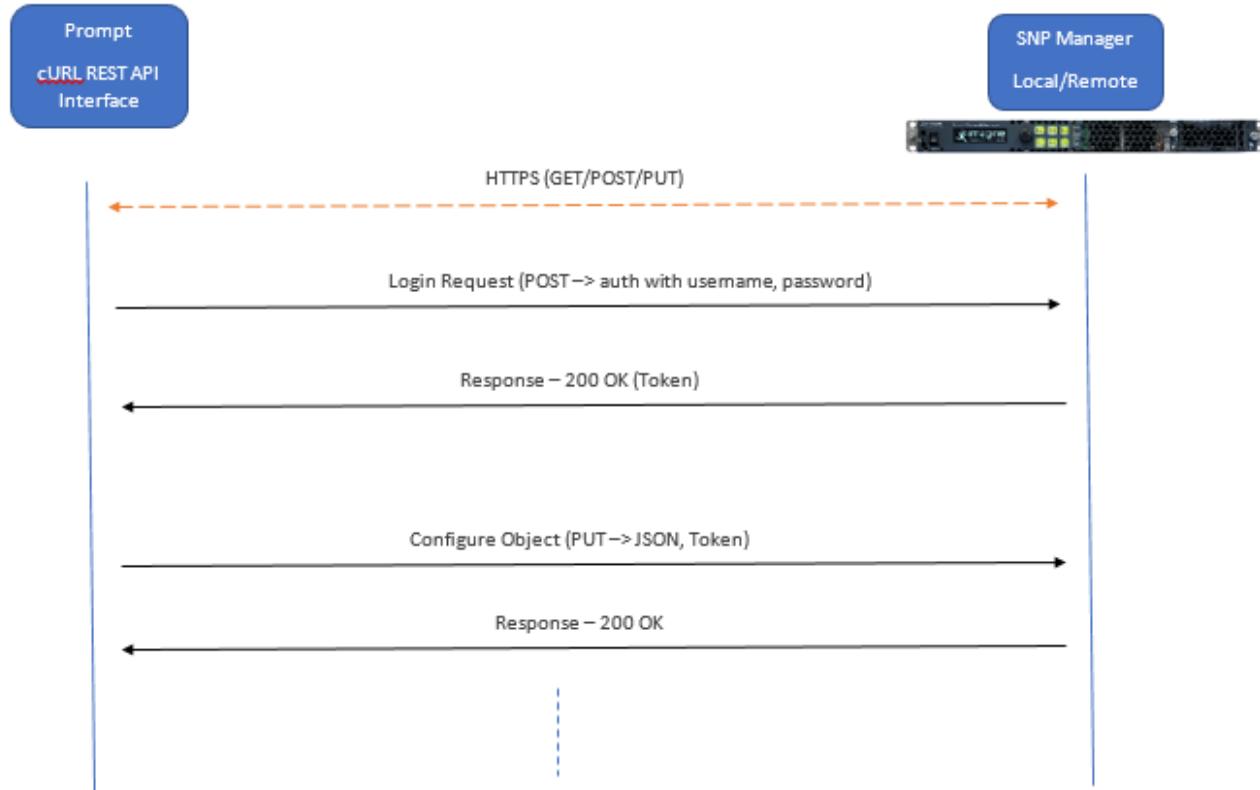
This publication is designed to assist in the use of the product as it exists on the date of publication of this manual and may not reflect the product at the current time or an unknown time in the future. This publication does not in any way warrant description accuracy or guarantee the use for the product to which it refers. Imagine Communications reserves the right, without notice to make such changes in equipment, design, specifications, components, or documentation as progress may warrant to improve the performance of the product.

- Summary
- Creating input JSON file
 - Identify target parameter(s)
 - Identify relevant JSON at SNP
 - Build JSON configuration for performing changing at SNP
- cURL Examples
 - Changing Processor Personality
 - Changing one SNP parameter e.g. TSG Pattern
 - Changing multiple SNP parameters e.g. License Tokens
 - Changing multiple SNP parameters e.g. Color Parameters
 - Example of switching between two SNP-MV layouts

Summary

In this article, we present some examples of SNP configuration changes using the Restful API using cURL commands. Click [here](#) for more information about "Client URL".

The sequence of communication between client and SNP restful API is as follows:

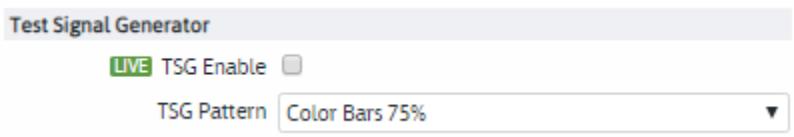


Creating input JSON file

JSON is used to encapsulate data that we send and/or receive when calling REST API. Below are some steps that can help in creating relevant JSON:

Identify target parameter(s)

In the snapshot from SNP UI below, we'll identify TSG Pattern as the target parameter:



We would change this parameter at Processor C, Section 2, which is having Conv personality.

Instead of changing the TSG Pattern by selecting the drop-down list from the UI, we'll use REST API to perform the change.

Identify relevant JSON at SNP

Steps below can be followed to identify relevant JSON that would help to create the partial JSON to be used for performing any changes to SNP.

1. SSH to SNP and login.
2. Change directory to `/mnt/ssd/opt/json`
3. Search for the label 'TSG Pattern' in this folder using grep. For example, grep 'TSG Pattern' *_schema.json.

For example:

```
SNP-2117480239:/mnt/ssd/opt/json$ grep 'TSG Pattern' *_schema.json

procChannelHD_schema.json:           "title": "TSG Pattern",

procChannelUHD_schema.json:          "title": "TSG Pattern",
```

4. Since Processor C is having 2160p output, we'll choose UHD JSON.

Build JSON configuration for performing changing at SNP

Open the JSON selected above and identify the following parameters/sections:

1. Type
2. Name
3. Object ID
4. Config

Take help from the examples below for better understanding.

cURL Examples

Changing Processor Personality

login.sh

```
curl -k -X POST -H "Content-Type:application/json" -d "{\"username\": \"admin\", \"password\": \"password\"}" https://172.10.100.119:9089/api/auth
```

pers.json - partial configuration

```
{
"config": {"type": "processor", "name": "processorC", "object_ID": "processorC", "general": {"personality": "Remap"}},
"type": "processor",
"name": "processorC",
"object_ID": "processorC"
}
```

Changing processor C personality to Remap mode.

change_pers.sh

```
curl -k -X PUT -H "Authorization:`./login.sh`" -H "Content-Type:application/json" -d@./pers.json https://172.10.100.119:9089/api/elements/172.10.100.119
/config/processorC
```

change_pers.sh script should be executed from prompt, it would call the login script to get the token and will also include the JSON.

Changing one SNP parameter e.g. TSG Pattern

tsg.json - partial configuration

```
{  
  "config": [{"type": "procChannelUHD", "name": "C-UHD-2", "object_ID": "C-UHD-2", "VideoTsg": {"Select": "Color Bars 75%"}},  
    {"type": "procChannelUHD",  
     "name": "C-UHD-2",  
     "object_ID": "C-UHD-2"}  
}
```

Changing TSG pattern to Color Bar 75% at Processor C, Section 2 and the mode is UHD.

change_tsg.sh

```
curl -k -X PUT -H "Authorization: `./login.sh`" -H "Content-Type:application/json" -d@./tsg.json https://172.10.100.119:9089/api/elements/172.10.100.119/config/C-UHD-2
```

change_tsg.sh script should be executed from prompt, it would call the login script to get the token and will also include the JSON.

Note: *login.sh can be reused from above example*

Changing multiple SNP parameters e.g. License Tokens

license.json

```
{  
  "config": [{"License_Tokens": [  
      {"procD": true, "procA": true, "procC": true, "procB": true, "keyName": "SNP-SK-QUAD-3GCONV", "keyTitle": "Quad 1080P Conversion Personality"},  
      {"procD": true, "procA": true, "procC": true, "procB": true, "keyName": "SNP-SK-DUAL-CONV", "keyTitle": "DUAL UHD Conversion personality"},  
      {"procD": true, "procA": true, "procC": true, "procB": true, "keyName": "SNP-SK-DUAL-MADI", "keyTitle": "+128 Audio/MADI Feature"},  
      {"procD": true, "procA": true, "procC": true, "procB": true, "keyName": "SNP-SK-HDR", "keyTitle": "HDR Adjustment Feature"},  
      {"procD": true, "procA": true, "procC": true, "procB": true, "keyName": "SNP-SK-MV", "keyTitle": "Multiviewer Personality"},  
      {"procD": true, "procA": true, "procC": true, "procB": true, "keyName": "SNP-SK-PROXHD", "keyTitle": "HD Downsample Feature"},  
      {"procD": true, "procA": true, "procC": true, "procB": true, "keyName": "SNP-SK-SYNC", "keyTitle": "SYNC/REMAP Personality"}],  
      {"type": "system", "name": "system", "object_ID": "system"},  
      {"type": "system",  
       "name": "system",  
       "object_ID": "system"}  
    ]}  
}
```

change_license.sh

```
curl -k -X PUT -H "Authorization: `./login.sh`" -H "Content-Type:application/json" -d@./license.json https://172.10.100.157:9089/api/elements/172.10.100.157/config/system
```

Note: *login.sh can be reused from above example*

Changing multiple SNP parameters e.g. Color Parameters

color.json

```
{  
  "config": [{"type": "procChannelUHD", "name": "C-UHD-2", "object_ID": "C-UHD-2", "ColorCorrector2": {"BlackLevel": 1, "CbOffset": 1, "CrOffset": 1, "Hue": 1, "LumaGain": 1, "ChromaGain": 1, "CbGain": 1, "CrGain": 1, "GOffset": 1, "BOffset": 1, "ROffset": 1, "GGain": 1, "BGain": 1, "RGain": 1}},  
    {"type": "procChannelUHD",  
     "name": "C-UHD-2",  
     "object_ID": "C-UHD-2"}  
}
```

change_color_inputs.sh

```
curl -k -X PUT -H "Authorization: `./login.sh`" -H "Content-Type:application/json" -d@./color.json https://172.10.100.119:9089/api/elements/172.10.100.119/config/C-UHD-2
```

Note: *login.sh can be reused from above example*

Example of switching between two SNP-MV layouts

layout1.json

```
{  
  "config": {"\"type\": \"processor\", \"name\": \"processorA\", \"object_ID\": \"processorA\", \"general\": {\"SelectLayoutDisplay2\": \"Layout 1PIP.xml\"}}},  
  \"type\": \"processor\",  
  \"name\": \"processorA\",  
  \"object_ID\": \"processorA\"\n}  
}
```

layout2.json

```
{  
  "config": {"\"type\": \"processor\", \"name\": \"processorA\", \"object_ID\": \"processorA\", \"general\": {\"SelectLayoutDisplay2\": \"Layout-9-PIPs.xml\"}}},  
  \"type\": \"processor\",  
  \"name\": \"processorA\",  
  \"object_ID\": \"processorA\"\n}  
}
```

change_layout1.sh

```
curl -k -X PUT -H "Authorization:`./login.sh`" -H "Content-Type:application/json" -d@./layout1.json https://172.10.100.119:9089/api/elements/172.10.100.119/config/processorA
```

change_layout2.sh

```
curl -k -X PUT -H "Authorization:`./login.sh`" -H "Content-Type:application/json" -d@./layout2.json https://172.10.100.119:9089/api/elements/172.10.100.119/config/processorA
```

Note: login.sh can be reused from above example

A small shell script can be used to switch between these 2 layouts, example is below:

```
switch_layouts.sh  
#!/bin/sh  
  
a=0  
while [ 1 ]  
do  
./change_layout1.sh  
sleep 30  
./change_layout2.sh  
sleep 30  
a=`expr $a + 1`  
echo -n "$a"  
done
```