

Command summary

HTTP operations are translated to commands depending on the URL:

API command	URL	Version
Get unit status	/sts/UnitStatus.json	
Get SRT status	/sts/SRTStatus.json	2.0
Get unit configuration	/config/UnitParamsList.json	
Get encoder configuration	/config/EncoderParamsList.json	
Get decoder configuration	/config/DecoderParamsList.json	
Get transmitter configuration	/config/TxParamsList.json	2.1
Get camera control configuration	/config/CamCtrlParamsList.json	3.0
Merge unit configuration	/cgi-bin/set_unit_val.cgi	
Merge encoder configuration	/cgi-bin/set_enc_val.cgi	
Merge decoder configuration	/cgi-bin/set_dec_val.cgi	
Merge transmitter configuration	/cgi-bin/set_tx_val.cgi	2.1
Merge camera control configuration	/cgi-bin/set_cam_ctrl_val.cgi	3.0
Run action	/cgi-bin/wui_action.cgi	

Data format

All request and response data will be in JSON format where applicable.

Requests which include data must include `Content-Type: application-json` in the header.

Requests must be one of the following data types:

- an object, e.g. `{}`

WARNING: There **must not** be a trailing comma after the last key/value in an object, or the last item in an array.

HTTP status codes

Successful commands will return 200 `Success`. Standard HTTP error codes will be used for invalid HTTP requests and redirections.

Python interface

An example python interface is provided in this documentation. The code here as a prerequisite for all over examples for API access.

Python code

```
import json
import requests
JSON_HEADERS = {'user-agent': 'sapp-agent', 'Content-type': 'application/json'}
ACTION_HEADERS = {'user-agent': 'sapp-agent'}
DEFAULT_GET_TIMEOUT = 1.0
DEFAULT_POST_TIMEOUT = 5.0
DEFAULT_ACTION_TIMEOUT = 10.0
```

Web commands

Get unit status

Description

Gets the complete status of the unit. This contains common items present in all units modes, some items are only present in one unit mode or under certain configuration conditions.

HTTP operation

GET

URL

<root-endpoint>/sts/UnitStatus.json

https://192.168.1.100/sts/UnitStatus.json

Request data

None.

Response data

A JSON object containing unit status.

```
{
  "UnitMode" : 0,
  "BoardType" : "B110"
}
```

Python code

```
def getUnitStatus(address):
    """
    Arguments:
        address (str): IP address to target the API request.
    Returns:
        JSON item (string, number, object or array) returned from API request.
    """
    url = "http://%s/sts/UnitStatus.json" % (address, )
    response = requests.request("GET", url, headers=JSON_HEADERS,
                               timeout=DEFAULT_GET_TIMEOUT)

    if response.status_code != 200:
        raise EnvironmentError("getUnitStatus failed: %u" % (response.status_code))
    return json.loads(response.text)
```

Get SRT status

Description

Gets the current status of the SRT connection. This will return 404 Not Found if streaming is not active on the unit.

HTTP operation

GET

URL

<root-endpoint>/sts/SRTStatus.json

https://192.168.1.100/sts/SRTStatus.json

Request data

None.

Response data

A JSON object containing SRT status.

```
{
    "SrtConnected" : 1,
    "SrtRemoteAddress" : "192.168.1.101"
}
```

Python code

```
def getSrtStatus(address):
    """
    Arguments:
        address (str): IP address to target the API request.
    Returns:
        JSON item (string, number, object or array) returned from API request.
    """
    url = "http://%s/sts/SRTStatus.json" % (address, )
    response = requests.request("GET", url, headers=JSON_HEADERS,
                               timeout=DEFAULT_GET_TIMEOUT)

    if response.status_code != 200:
        raise EnvironmentError("getSrtStatus failed: %u" % (response.status_code))
    return json.loads(response.text)
```

Get unit configuration

Description

Gets the current unit configuration.

HTTP operation

GET

URL

<root-endpoint>/config/UnitParamsList.json

https://192.168.1.100/config/UnitParamsList.json

Request data

None.

Response data

A JSON object containing unit configuration.

```
{
    "UnitMode" : 0,
    "UnitName" : "Sapphire"
}
```

Python code

```
def getUnitConfig(address):
    """
    Arguments:
        address (str): IP address to target the API request.
    Returns:
        JSON item (string, number, object or array) returned from API request.
    """
    url = "http://%s/config/UnitParamsList.json" % (address, )
    response = requests.request("GET", url, headers=JSON_HEADERS,
                               timeout=DEFAULT_GET_TIMEOUT)
```

```
if response.status_code != 200:
    raise EnvironmentError("getUnitConfig failed: %u" % (response.status_code))
return json.loads(response.text)
```

Get encoder configuration

Description

Gets the current encoder configuration. This might return 404 Not Found if encoder functionality is not available on the unit.

HTTP operation

GET

URL

<root-endpoint>/config/EncoderParamsList.json

https://192.168.1.100/config/EncoderParamsList.json

Request data

None.

Response data

A JSON object containing encoder configuration.

```
{
  "ProgramNumber" : 1,
  "ServiceName" : "Service 1"
}
```

Python code

```
def getEncoderConfig(address):
    """
    Arguments:
        address (str): IP address to target the API request.
    Returns:
        JSON item (string, number, object or array) returned from API request.
    """
    url = "http://%s/config/EncoderParamsList.json" % (address, )
    response = requests.request("GET", url, headers=JSON_HEADERS,
                               timeout=DEFAULT_GET_TIMEOUT)

    if response.status_code != 200:
        raise EnvironmentError("getEncoderConfig failed: %u" % (response.status_code))
    return json.loads(response.text)
```

Get decoder configuration

Description

Gets the current decoder configuration. This might return 404 Not Found if decoder functionality is not available on the unit.

HTTP operation

GET

URL

<root-endpoint>/config/DecoderParamsList.json

https://192.168.1.100/config/DecoderParamsList.json

Request data

None.

Response data

A JSON object containing decoder configuration.

```
{
  "ProgramNumber" : 1,
  "DefServiceName" : "Service 1"
}
```

Python code

```
def getDecoderConfig(address):
    """
    Arguments:
        address (str): IP address to target the API request.
    Returns:
        JSON item (string, number, object or array) returned from API request.
    """
    url = "http://%s/config/DecoderParamsList.json" % (address, )
    response = requests.request("GET", url, headers=JSON_HEADERS,
                               timeout=DEFAULT_GET_TIMEOUT)

    if response.status_code != 200:
        raise EnvironmentError("getDecoderConfig failed: %u" % (response.status_code))
    return json.loads(response.text)
```

Get transmitter configuration

Description

Gets the current transmitter configuration. This might return 404 Not Found if transmitter functionality is not available on the unit.

HTTP operation

GET

URL

<root-endpoint>/config/TxParamsList.json

https://192.168.1.100/config/TxParamsList.json

Request data

None.

Response data

A JSON object containing transmitter configuration.

```
{
  "mod_output" : 1,
  "mod_freq" : "2265.0"
}
```

Python code

```
def getTransmitterConfig(address):
    """
    Arguments:
```

```

    address (str): IP address to target the API request.
Returns:
    JSON item (string, number, object or array) returned from API request.
"""
url = "http://%s/config/TxParamsList.json" % (address, )
response = requests.request("GET", url, headers=JSON_HEADERS,
                            timeout=DEFAULT_GET_TIMEOUT)

if response.status_code != 200:
    raise EnvironmentError("getTransmitterConfig failed: %u" % (response.status_code))
return json.loads(response.text)

```

Get camera control configuration

Description

Gets the current camera control configuration. This might return 404 Not Found if camera control functionality is not available on the unit.

HTTP operation

GET

URL

<root-endpoint>/config/CamCtrlParamsList.json
 https://192.168.1.100/config/CamCtrlParamsList.json

Request data

None.

Response data

A JSON object containing camera control configuration.

```

{
  "cam_type" : 1,
  "cam_number" : 1
}

```

Python code

```

def getCameraControlConfig(address):
    """
    Arguments:
        address (str): IP address to target the API request.
    Returns:
        JSON item (string, number, object or array) returned from API request.
    """
    url = "http://%s/config/CamCtrlParamsList.json" % (address, )
    response = requests.request("GET", url, headers=JSON_HEADERS,
                                timeout=DEFAULT_GET_TIMEOUT)

    if response.status_code != 200:
        raise EnvironmentError("getCameraControlConfig failed: %u" % (response.status_code))
    return json.loads(response.text)

```

Merge unit configuration

Description

Update the unit configuration by merging provided parameters with the current configuration. When merging an object, any items which aren't provided will left as their previous values.

HTTP operation

POST

URL

<root-endpoint>/cgi-bin/set_unit_val.cgi

https://192.168.1.100/cgi-bin/set_unit_val.cgi

Request data

A JSON item (object) containing the configuration to be merged.

```
{
  "UnitMode" : 0,
  "UnitName" : "Sapphire"
}
```

Response data

None.

Python code

```
def mergeUnitConfig(address, config={}):
    """
    Arguments:
        address (str): IP address to target the API request.
        config (object): JSON object to merge.
    Returns:
        None.
    """
    url = "http://%s/cgi-bin/set_unit_val.cgi" % (address, )
    response = requests.request("POST", url, headers=JSON_HEADERS, json=config,
                                timeout=DEFAULT_POST_TIMEOUT)

    if response.status_code != 200:
        raise EnvironmentError("mergeUnitConfig failed: %u" % (response.status_code))
```

Merge encoder configuration

Description

Update the encoder configuration by merging provided parameters with the current configuration. When merging an object, any items which aren't provided will left as their previous values.

HTTP operation

POST

URL

<root-endpoint>/cgi-bin/set_enc_val.cgi

https://192.168.1.100/cgi-bin/set_enc_val.cgi

Request data

A JSON item (object) containing the configuration to be merged.

```
{
  "ProgramNumber" : 1,
  "ServiceName" : "Service 1"
}
```

Response data

None.

Python code

```
def mergeEncoderConfig(address, config={}):
    """
    Arguments:
        address (str): IP address to target the API request.
        config (object): JSON object to merge.
    Returns:
        None.
    """
    url = "http://%s/cgi-bin/set_enc_val.cgi" % (address, )
    response = requests.request("POST", url, headers=JSON_HEADERS, json=config,
                               timeout=DEFAULT_POST_TIMEOUT)

    if response.status_code != 200:
        raise EnvironmentError("mergeEncoderConfig failed: %u" % (response.status_code))
```

Merge decoder configuration

Description

Update the decoder configuration by merging provided parameters with the current configuration. When merging an object, any items which aren't provided will left as their previous values.

HTTP operation

POST

URL

<root-endpoint>/cgi-bin/set_dec_val.cgi

https://192.168.1.100/cgi-bin/set_dec_val.cgi

Request data

A JSON item (object) containing the configuration to be merged.

```
{
  "ProgramNumber" : 1,
  "DefServiceName" : "Service 1"
}
```

Response data

None.

Python code

```
def mergeDecoderConfig(address, config={}):
    """
    Arguments:
        address (str): IP address to target the API request.
        config (object): JSON object to merge.
    Returns:
        None.
    """
    url = "http://%s/cgi-bin/set_dec_val.cgi" % (address, )
    response = requests.request("POST", url, headers=JSON_HEADERS, json=config,
                               timeout=DEFAULT_POST_TIMEOUT)
```

```
if response.status_code != 200:
    raise EnvironmentError("mergeDecoderConfig failed: %u" % (response.status_code))
```

Merge transmitter configuration

Description

Update the transmitter configuration by merging provided parameters with the current configuration. When merging an object, any items which aren't provided will left as their previous values.

HTTP operation

POST

URL

<root-endpoint>/cgi-bin/set_tx_val.cgi

https://192.168.1.100//cgi-bin/set_tx_val.cgi

Request data

A JSON item (object) containing the configuration to be merged.

```
{
  "mod_output" : 1,
  "mod_freq" : "2265.0"
}
```

Response data

None.

Python code

```
def mergeTransmitterConfig(address, config={}):
    """
    Arguments:
        address (str): IP address to target the API request.
        config (object): JSON object to merge.
    Returns:
        None.
    """
    url = "http://%s/cgi-bin/set_tx_val.cgi" % (address, )
    response = requests.request("POST", url, headers=JSON_HEADERS, json=config,
                               timeout=DEFAULT_POST_TIMEOUT)

    if response.status_code != 200:
        raise EnvironmentError("mergeTransmitterConfig failed: %u" % (response.status_code))
```

Merge camera control configuration

Description

Update the camera control configuration by merging provided parameters with the current configuration. When merging an object, any items which aren't provided will left as their previous values.

HTTP operation

POST

URL

<root-endpoint>/cgi-bin/set_cam_ctrl_val.cgi
https://192.168.1.100//cgi-bin/set_cam_ctrl_val.cgi

Request data

A JSON item (object) containing the configuration to be merged.

```
{  
  "cam_type" : 1,  
  "cam_number" : 1  
}
```

Response data

None.

Python code

```
def mergeCameraControlConfig(address, config={}):  
    """  
    Arguments:  
        address (str): IP address to target the API request.  
        config (object): JSON object to merge.  
    Returns:  
        None.  
    """  
    url = "http://%s/cgi-bin/set_cam_ctrl_val.cgi" % (address, )  
    response = requests.request("POST", url, headers=JSON_HEADERS, json=config,  
                               timeout=DEFAULT_POST_TIMEOUT)  
  
    if response.status_code != 200:  
        raise EnvironmentError("mergeCameraControlConfig failed: %u" % (response.status_code))
```

Run action

Description

Triggers a specific one shot action on the unit.

HTTP operation

POST

URL

<root-endpoint>/cgi-bin/wui_action.cgi
https://192.168.1.100//cgi-bin/wui_action.cgi

Request data

A JSON item (object) containing the action and any additional parameters required for that action.

```
{  
  "action" : "6",  
  "fs" : "vfat"  
}
```

Response data

Raw text response with message.

Python code

```
def runAction(address, data={}):
    """
    Arguments:
        address (str): IP address to target the API request.
        data (object): JSON object containing action parameters.
    Returns:
        None.
    """
    url = "http://%s/cgi-bin/wui_action.cgi" % (address, )
    response = requests.request("POST", url, headers=ACTION_HEADERS, data=data,
                               timeout=DEFAULT_ACTION_TIMEOUT)

    if response.status_code != 200:
        raise EnvironmentError("runAction failed: %u" % (response.status_code))
    return response.text
```