RestAPI Definition

Change History

Date	Description
April 6th 2020	First edition
June 5th 2020	Introduction of uuids
June 30th 2020	Minor corrections and more information about error handling
July 20th 2020	Minor corrections regarding invalid argument error handling

Introduction

The Media Suite RestAPI defines some basic controls of a Vision Mixer that can be controlled from an automation system that needs to control a Media Suite device. The following document lists all the types and commands provided by the API.

The interface of the Media Suite RestAPI requires the following command syntax

```
http(s)://<ip>(:<port>)/<json-pointer>
```

Note: trailing backslashes are not allowed and will produce a 404 Invalid url response.

Object Types

Base

Those parameters are inherited by all types of objects.

name	R/W	comments
uuid	R	E.g. **93ddd7c7-752f-505f-8e62-444ffc86428c**

AUX

name	R/W	comments
name	R	
source	R/W	
sources	R	

Macro

name	R/W	comments
name	R	
state	W	null if read

Input

name	R/W	comments
index	R	
name	R	

name	R/W	comments
tally	R	

Scene

name	R/W	comments
name	R	
tally	R	
layers	R	

Layer

name	R/W	comments
name	R	
sourceA	R/W	optional
sourceB	R/W	optional
sources	R	

Multiviewer

name	R/W	comments
index	R	
name	R	
preset	W	null if read
presets	R	
sdp	R	

Multiviewer Preset

name	R/W	comments
id	R	
name	R	
usr	R	

Object identification

The initial way to identify objects was by using their name attribute or an index value where applicable. This is an easy way to identify or name objects if some user wants to execute commands directly from a browser or by using curl. The drawback of this is mostly when someone want to address object types that are created dynamically like scenes. The name of the scene is defined by the user and it is allowed to have duplicate names. In such a case it is not possible to address a specfic object because the name itself is not unique to the object. To solve this problem the unique identifiers where introduced. Even though it is harder to use from a browser or via curl for an application developer this is the recommended way to address objects.

Note: To have the uuids available for example for scene objects a /scenes query needs to be setup first. The uuids in response can be used as long as they are valid. Right now the uuids are not automatically generated if for example a new scene appears on the system. This will be added in a later version of the protocol.

GET request

The get requests supported by the MediaSuite RestAPI are mostly following the RFC7231. One exception is the /command interface that will be described later.

GET Inputs

/inputs


```
[
    {
        "index": 0,
        "name": "IN1",
        "tally": 1,
        "uuid": "e53210f7-2235-5ae3-9c02-4f58d67bf8b8"
    },
    {
        "index": 0,
        "name": "IN2",
        "tally": 0,
        "uuid": "bc2932d5-bc00-52a3-be8b-753532420c14"
    },
    {
        "index": 0,
        "name": "IN3",
        "tally": 0,
        "uuid": "39ca682a-1adc-573e-a702-a53314dc81dd"
    },
    {
        "index": 0,
        "name": "IN4",
        "tally": 0,
        "uuid": "efab3a92-1fc5-55ef-bd0c-91b12b17b8e1"
    },
    {
        "index": 0,
        "name": "IN5",
        "tally": 0,
        "uuid": "3b4e76eb-e6a8-5290-828b-e1659cec1dd4"
    },
    {
        "index": 0,
        "name": "IN6",
        "tally": 0,
        "uuid": "dbd5ae62-5944-5de6-a44f-9afe764e28ff"
    },
    {
        "index": 0,
        "name": "IN7",
        "tally": 0,
        "uuid": "eb1564a8-c2f2-5c96-8e5e-34a98637c8b5"
    },
    {
        "index": 0,
        "name": "IN8",
        "tally": 0,
        "uuid": "6b977973-ea4a-578b-8324-4248f29f33ea"
    },
    {
        "index": 0,
        "name": "IN9",
        "tally": 0,
        "uuid": "e41935e9-8dea-5270-b58c-70d35dc5c949"
    },
    {
        "index": 0,
        "name": "IN10",
        "tally": 0,
```

```
        "uuid": "07ea9b8b-ac21-52b9-8ba6-a859085890cc"
    },
    {
        "index": 0,
        "name": "IN11",
        "tally": 0,
        "uuid": "51b21378-42be-5703-9efe-cd2baf5719f7"
    },
    {
        "index": 0,
        "name": "IN12",
        "tally": 0,
        "uuid": "188bfc2f-83ed-5543-b8d9-bc5f6da82d24"
    },
    {
        "index": 0,
        "name": "IN13",
        "tally": 0,
        "uuid": "e0a05c4a-8bc9-5fe4-8db9-92167c4ff611"
    },
    {
        "index": 0,
        "name": "IN14",
        "tally": 0,
        "uuid": "59bfff32-9f79-5334-8bdd-87b7c3f5ff7b"
    },
    {
        "index": 0,
        "name": "IN15",
        "tally": 0,
        "uuid": "46b00464-4e04-5e84-be16-7a688f1f3f66"
    },
    {
        "index": 0,
        "name": "IN16",
        "tally": 0,
        "uuid": "7e5fc323-ceb7-5fef-a5ae-0ae85ed6cc97"
    },
    {
        "index": 0,
        "name": "IN17",
        "tally": 0,
        "uuid": "2c961af2-ed44-5dd3-91ef-72c5dcda076a"
    },
    {
        "index": 0,
        "name": "IN18",
        "tally": 0,
        "uuid": "dd7a9779-b511-551e-bbdd-7f52907261a3"
    },
    {
        "index": 0,
        "name": "IN19",
        "tally": 0,
        "uuid": "be7cf1dc-b5c6-5412-a87a-4d3e6d3b850e"
    },
    {
        "index": 0,
        "name": "IN20",
        "tally": 0,
        "uuid": "8eb2f82f-d45f-5d7b-831f-8e6ca06beda8"
    },
    {
        "index": 0,
        "name": "IN21",
        "tally": 0,
```

```
        "tally": 0,
        "uuid": "5ed6992c-7b1b-52d7-91c4-f2aeb76410e8"
    },
    {
        "index": 0,
        "name": "IN22",
        "tally": 0,
        "uuid": "d904e11f-2e72-5adb-bb1a-3563ee620b2d"
    },
    {
        "index": 0,
        "name": "IN23",
        "tally": 0,
        "uuid": "2ba9ef85-02b1-54aa-9156-51b58bd1e176"
    },
    {
        "index": 0,
        "name": "IN24",
        "tally": 0,
        "uuid": "cee82917-ed52-5431-adf1-64c9b5d977d8"
    },
    {
        "index": 0,
        "name": "IN25",
        "tally": 0,
        "uuid": "ed708724-084e-5e73-93fa-9be4b12293ee"
    },
    {
        "index": 0,
        "name": "IN26",
        "tally": 0,
        "uuid": "875b2a5f-a8e5-5ea3-9ec3-33432d6df1d1"
    },
    {
        "index": 0,
        "name": "IN27",
        "tally": 0,
        "uuid": "43cd1c09-863b-5fa4-aff7-1c47bce225be"
    },
    {
        "index": 0,
        "name": "IN28",
        "tally": 0,
        "uuid": "679e335b-94c4-5d87-b451-57e600c88c53"
    },
    {
        "index": 0,
        "name": "IN29",
        "tally": 0,
        "uuid": "0ebf463e-6eed-5a3a-88b1-2ed1ca9296c6"
    },
    {
        "index": 0,
        "name": "IN30",
        "tally": 0,
        "uuid": "f3f86a30-b573-509f-bdce-6e17e72a3b7d"
    },
    {
        "index": 0,
        "name": "IN31",
        "tally": 0,
        "uuid": "42487096-8934-500e-83c5-31efbce18499"
    },
    {
        "index": 0,
        "name": "IN32",
```

```
        "tally": 0,
        "uuid": "812135dc-7863-5ab3-b182-7db4affcea02"
    }
]
```

/input/\<input> or /inputs/0 or /inputs/\<uuid>

```
{
    "index": 0,
    "name": "IN1",
    "tally": 1,
    "uuid": "e53210f7-2235-5ae3-9c02-4f58d67bf8b8"
}
```

GET Macros

/macros

```
[
    {
        "name": "Macro1",
        "state": null,
        "uuid": "eb71d141-181c-5390-b709-7a9d50789bb1"
    },
    {
        "name": "Macro2",
        "state": null,
        "uuid": "3693f3ff-9b0b-5377-9090-a73e4fe58207"
    }
]
```

/macros/\<macro> or /macros/\<uuid>

```
{
    "name": "Macro1",
    "state": null,
    "uuid": "eb71d141-181c-5390-b709-7a9d50789bb1"
}
 ````

GET AUX
Note: The following responses contain source options "sources", a list of elements which is reduced in this example.

/aux
````json
[
    {
        "name": "AUX1",
        "source": "Main",
        "sources": [
            "Black",
            "White"
        ],
        "uuid": "04c03dbe-aa4e-5bdd-a98d-942f5c19ecbd"
    },
    {
        "name": "AUX2",
        "source": "Main",
        "sources": [
            "Black",
            "White"
        ],
        "uuid": "2ce35df9-f935-5fb3-b48a-1ab7ead6f660"
```

```
        "uuid": "2ce55df9-f955-5fb5-b48a-1ab7ead0f660"
    },
    {
        "name": "AUX3",
        "source": "Black",
        "sources": [
            "Black",
            "White"
        ],
        "uuid": "96a28707-4daf-5d2d-af33-b90fa6bfeb48"
    },
    {
        "name": "AUX4",
        "source": "Black",
        "sources": [
            "Black",
            "White"
        ],
        "uuid": "5ed14a51-cbb3-53ee-8b17-80ac6c6f502d"
    },
    {
        "name": "AUX5",
        "source": "Black",
        "sources": [
            "Black",
            "White"
        ],
        "uuid": "711031aa-e0ba-505b-a458-972db61d2991"
    },
    {
        "name": "AUX6",
        "source": "Black",
        "sources": [
            "Black",
            "White"
        ],
        "uuid": "c8861948-d378-513f-88e1-c591fd4c39df"
    },
    {
        "name": "AUX7",
        "source": "Black",
        "sources": [
            "Black",
            "White"
        ],
        "uuid": "1f535060-ac03-5802-b4ff-846037b3cb52"
    },
    {
        "name": "AUX8",
        "source": "Black",
        "sources": [
            "Black",
            "White"
        ],
        "uuid": "482c1e75-0129-5b80-84e8-e2d1cd4f0e55"
    },
    {
        "name": "AUX9",
        "source": "Black",
        "sources": [
            "Black",
            "White"
        ],
        "uuid": "74c46861-f517-5c92-be44-d24b0a8fe93a"
    },
    {
```

```
        "name": "AUX10",
        "source": "Black",
        "sources": [
            "Black",
            "White"
        ],
        "uuid": "95f9a11f-697b-5831-a9c1-9797ad60df1d"
    },
    {
        "name": "AUX11",
        "source": "Black",
        "sources": [
            "Black",
            "White"
        ],
        "uuid": "a5602a21-6d3d-549a-82a5-d7d0f5af6be6"
    },
    {
        "name": "AUX12",
        "source": "Black",
        "sources": [
            "Black",
            "White"
        ],
        "uuid": "94c58c55-363b-5fa8-902a-4e29825af438"
    },
    {
        "name": "AUX13",
        "source": "Black",
        "sources": [
            "Black",
            "White"
        ],
        "uuid": "783d0542-978f-51c2-8a98-5e2607763e9a"
    },
    {
        "name": "AUX14",
        "source": "Black",
        "sources": [
            "Black",
            "White"
        ],
        "uuid": "c3f81795-c004-59a0-b388-879af43c5f16"
    },
    {
        "name": "AUX15",
        "source": "Black",
        "sources": [
            "Black",
            "White"
        ],
        "uuid": "0dc7d638-af96-53eb-802e-222cc65b5681"
    },
    {
        "name": "AUX16",
        "source": "Black",
        "sources": [
            "Black",
            "White"
        ],
        "uuid": "06f3d0c0-20ea-5c90-b479-4f49d95d26c5"
    },
    {
        "name": "AUX17",
        "source": "Black",
```

```
        "sources": [
            "Black",
            "White"
        ],
        "uuid": "15f8570d-984f-546c-822f-5e0eceb18af3"
    },
    {
        "name": "AUX18",
        "source": "Black",
        "sources": [
            "Black",
            "White"
        ],
        "uuid": "45a00e60-e7a8-559c-b5c6-7a21280ae780"
    },
    {
        "name": "AUX19",
        "source": "Black",
        "sources": [
            "Black",
            "White"
        ],
        "uuid": "318b2e8b-3bf2-5646-8455-97925faab729"
    },
    {
        "name": "AUX20",
        "source": "Black",
        "sources": [
            "Black",
            "White"
        ],
        "uuid": "d397acb3-349b-5c48-8327-e43ff3610a01"
    }
]
```

#### /aux/0 or /aux/AUX1 or /aux/\<uuid>

```
{
    "name": "AUX1",
    "source": "Main",
    "sources": [
        "Black",
        "White"
    ],
    "uuid": "04c03dbe-aa4e-5bdd-a98d-942f5c19ecbd"

}
```

## GET Multiviewers

#### /multiviewers

```json [ { "index": 0, "name": "Multiviewer1", "preset": null, "presets": [ { "id": 0, "name": "Full", "usr": false }, { "id": 1, "name": "Quad Split", "usr": false }, { "id": 2, "name": "10 Split A", "usr": false }, { "id": 3, "name": "10 Split B", "usr": false }, { "id": 4, "name": "10 Split C", "usr": false }, { "id": 5, "name": "10 Split D", "usr": false }, { "id": 6, "name": "9 Split", "usr": false }, { "id": 7, "name": "16 Split", "usr": false }, { "id": 8, "name": "25 Split", "usr": false }, { "id": 9, "name": "36 Split", "usr": false } ], "sdp": "v=0\nc=IN IP4 239.168.50.115\ns=Multiviewer 0\nt=0 0\nm=video 20000 RTP/AVP 96\na=rtpmap:96 H264/90000\n", "uuid": "fd839e94-a9e9-570b-a5aa-bf99575f364f" }, { "index": 1, "name": "Multiviewer2", "preset": null, "presets": [ { "id": 0, "name": "Full", "usr": false }, { "id": 1, "name": "Quad Split", "usr": false }, { "id": 2, "name": "10 Split A", "usr": false }, { "id": 3, "name": "10 Split B", "usr": false }, { "id": 4, "name": "10 Split C", "usr": false }, { "id": 5, "name": "10 Split D", "usr": false }, { "id": 6, "name": "9 Split", "usr": false }, { "id": 7, "name": "16 Split", "usr": false }, { "id": 8, "name": "25 Split", "usr": false }, { "id": 9, "name": "36 Split", "usr": false } ], "sdp": "v=0\nc=IN IP4 239.168.50.115\ns=Multiviewer 1\nt=0 0\nm=video 20001 RTP/AVP 96\na=rtpmap:96 H264/90000\n", "uuid": "ef884504-d688-59c7-a2a0-2b7d69e70e5b" } ]

```
#### /multiviewers/0 or /multiviewers/\<multiviewer\> or /multiviewers/\<uuid\>
````json
{
    "index": 0,
```

```
    "name": "Multiviewer1",
    "preset": null,
    "presets": [
        {
            "id": 0,
            "name": "Full",
            "usr": false
        },
        {
            "id": 1,
            "name": "Quad Split",
            "usr": false
        },
        {
            "id": 2,
            "name": "10 Split A",
            "usr": false
        },
        {
            "id": 3,
            "name": "10 Split B",
            "usr": false
        },
        {
            "id": 4,
            "name": "10 Split C",
            "usr": false
        },
        {
            "id": 5,
            "name": "10 Split D",
            "usr": false
        },
        {
            "id": 6,
            "name": "9 Split",
            "usr": false
        },
        {
            "id": 7,
            "name": "16 Split",
            "usr": false
        },
        {
            "id": 8,
            "name": "25 Split",
            "usr": false
        },
        {
            "id": 9,
            "name": "36 Split",
            "usr": false
        }
    ],
    "sdp": "v=0\nc=IN IP4 239.168.10.209
            s=Multiviewer 0
            t=0 0
            m=video 20000 RTP/AVP 96
            a=rtpmap:96 H264/90000",
    "uuid": "fd839e94-a9e9-570b-a5aa-bf99575f364f"


}
```


**/multiviewers/0/sdp or /multiviewers/\<uuid>/sdp**

If the sdp parameter is requested directly the response sends it with the Content-Type "application/sdp" as defined in RFC4566.

```
v=0
c=IN IP4 239.168.10.209
s=Multiviewer 0
t=0 0
m=video 20000 RTP/AVP 96
a=rtpmap:96 H264/90000
```

## GET Scenes

### /scenes


```
[
    {
        "layers": [
            {
                "name": "Background",
                "sourceA": "Black",
                "sourceB": "Black",
                "sources": [
                    "Black"
                ],
                "uuid": "ce412da9-e248-567a-a8e0-2ec65739623c"
            },
            {
                "name": "Layer-1",
                "sourceA": "Black",
                "sources": [
                    "Black",
                    "White"
                ],
                "uuid": "61508e94-fefb-5892-99e1-9c327e2149d6"
            },
            {
                "name": "Layer-2",
                "sourceA": "Black",
                "sources": [
                    "Black",
                    "White"
                ],
                "uuid": "93ddd7c7-752f-505f-8e62-444ffc86428c"
            }
        ],
        "name": "Main",
        "tally": 1,
        "uuid": "f1a5c030-e1a3-5b01-97e0-8a18b16ab1d3"
    }
]
```


### /scenes/\<scene> or /scenes/\<uuid>

```
{
    "layers": [
        {
            "name": "Background",
            "sourceA": "Black",
            "sourceB": "Black",
            "sources": [
                "Black"
            ],
            "uuid": "ce412da9-e248-567a-a8e0-2ec65739623c"
        },
        {
            "name": "Layer-1",
            "sourceA": "Black",
            "sources": [
                "Black",
                "White"
            ],
            "uuid": "61508e94-fefb-5892-99e1-9c327e2149d6"
        },
        {
            "name": "Layer-2",
            "sourceA": "Black",
            "sources": [
                "Black",
                "White"
            ],
            "uuid": "93ddd7c7-752f-505f-8e62-444ffc86428c"
        }
    ],
    "name": "Main",
    "tally": 1,
    "uuid": "f1a5c030-e1a3-5b01-97e0-8a18b16ab1d3"
}
```


#### /scenes/\<scene>/\<layer>

NOT YET IMPLEMENTED

# PATCH requests

The patch requests supported by the MediaSuite RestAPI are application/merge-patch+json types as described in RFC7396. Some of the definitions are not implemented direclty as described in the rfc because of system restrictions that prevent some command execution. For example, it is not possible to add/remove parameters to an object, because the data structures are fixed and not modifyable. This definition makes patch requests of the type application/json-patch+json as described in RFC6902 not applicable.

If a patch request is issued that contains invalid parameters it is not reported as an error, such parameters are ignored. For example a patch request with two parameters, one of them unknown to the system, the unknown parameter is ignored, the other parameter patch is executed if possible.

## PATCH scene

The scene json object contains a name and a tally paramters as well as a list of layers. All of them are read only properties. A patch request can be send by specifying new Layer parameters "sourceA", "sourceB". The allowed values are listed in the sources parameter. The sources cannot be changed from the RestAPI. This list of sources is provided by the application and can be specified via other tools like the MediaSuite GUI.

An example PATCH request that changes the Background Layer (sourceA) of the Main Scene can be written like this.

## PATCH /scenes/\<scene>/\<layer> or /scenes/\<uuid>/\<uuid>

Note: the source parameter is only accepted if the value is represented in the source options "sources" list. If the item is not listed in the source options the response is a 400 bad request status code.

TODO: replace status code 400 with 422 - Unprocessable entity!


```
{
    "sourceA": "Black",
    "sourceB": "White"
}
```

### PATCH /macros/\ or /macros/\<uuid>

```
{
    "state": "play"
}
```

### PATCH /aux/\ or /aux/\<uuid>

Note: the source parameter is only accepted if the value is represented in the source options "sources" list. If the item is not listed in the source options the response is a 400 bad request status code.

TODO: replace status code 400 with 422 - Unprocessable entity!

```
{
    "source": "Black"
}
```

### PATCH /multiviewers/\ or /multiviewers/\<uuid>

Note: the preset parameter is only accepted if the value is represented in the multiviewer presets list. If the item is not listed in the multiviewer presets list the response is a 400 bad request status code.

TODO: replace status code 400 with 422 - Unprocessable entity!

```
{
    "preset": 11
}
```

It is also possible to send the entire definition of the Scene, with just one parameter changed, back to the api. This is usually done in a PUT request, which is not supported because it would require a resend of the source option parameters for no reason.

## Other requests

In the current state of the MediaSuite Rest API PUT/DELETE/POST requests are not supported. If such a request is issues the System will send a 400 Bad Request status message.

## Error Handling

### Invalid Arguments in request body

The handling of patch request require a json document as request body that contain parameters that need to be changed. In case of invalid arguments the request is not accepted even if there are valid parameters listed. Invalid arguments are arguments that are not part of the object description received from the Rest API. This allows a client to receive an object description, update the desired parameter and send the entire object with the changes applied back to the server. Some of the arguments provided are read only arguments, if the client changes these arguments, these changes are ignored by the server. Optional parameters exist for layer elements. In this case it is required to specify at least one of the source parameters (sourceA, sourceB). A patch request like /aux/04c03dbe-aa4e-5bdd-a98d-942f5c19ecbd with the invalid request body

```
{
    "source": "Black",
    "foo": "bar"
}
```

responds with status code 400 Bad request and the change on the source parameter is not executed.

### Malformed json in request body

If a patch request is send the request body json document is parsed. In case of an error in the json format the system responds with a http status code 400 Bad request and a error message that describes the json document error. An example error response for the incorrect json document

```
{
     "A": "B
}
```

looks like this:

```
{
    "code": 400,
    "text": "Bad request. Invalid json format: * Line 2, Column 11 Syntax error: Malformed object literal"
}
```

The error message shows that in Line 2 Column 11 the \" character is missing which causes an Malformed object literal that cannot properly processed by the json parser.

Note: The Column number can be different if tabs are used instead of space characters.

# Command Interface

The following chapter shows a different kind of interface that is able to fullfill all the mentioned operations in the previous chapters. The described GET and PATCH request schemes following the rules of restful apis. Some legacy systems are using HTTP APIs that are completly focused on GET requests. This allows the user to specify a command like a patch requst directly in the address bar of the browser if the query arguments are know.

## Translation

To allow an identical behavior of the rest api and the command interface and translation unit is implemented that operates on the /command node and translates it into the corresponding PATCH request.

A GET /command requests therefore requires a query argument "object" that identifies the patchable node. Additionally a set of key=value query arguments can be passed which are used to form the PATCH request body.

Note: In the current state of the system (April 6th 2020) it is not possible to use the /command interface to patch an entire hierarchy structure.

### Example

The following command interface request

```
/command?object=/scenes/Main/Background&sourceA:Black&sourceB=White
```

will be translated into

```
PATCH /scenes/Main/Background
{
    "sourceA": "Black",
    "sourceB": "White"
}
```