JSON RPC for Encore3 Processors v10.0

General

JSON (JavaScript Object Notation) is a lightweight format that is used for interchanging data. It is based on a subset of JavaScript language: the way objects are built in JavaScript.

Introduction to JSON

JSON is built on two structures:

- A collection of name/value pairs: In various languages, this is realized as an object, record, struct, dictionary, hash table, keyed list, or associative array.
- An ordered list of values: In most languages, this is realized as an array, vector, list, or sequence.

Here is an example of JSON data:

```
{
     "firstName": "John",
     "lastName": "Smith",
     "address": {
         "streetAddress": "21 2nd Street",
         "city": "New York",
         "state": "NY",
         "postalCode": 10021
     },
     "phoneNumbers": [
         "212 555-1234",
         "646 555-4567"
     ]
 }
```

How JSON interacts with Event Master processors

JSON uses JSON RPC (REST based) to interact with the Event Master processors such as Encore3, E2, S3–4K, and EX. JSON-RPC is a remote procedure call protocol encoded in JSON.

JSON-RPC works by sending a request to a server implementing this protocol. The client in that case is typically software intending to call a single method of a remote system. Multiple input parameters can be passed to the remote method as an array or object, whereas the method itself can return multiple output data as well.

There are JSON RPCs defined to perform tasks on the Event Master system. The user needs to send JSON requests through their application or open-source application like Postman.

These applications should send non-secure requests on the IP address where the Event Master host processor is running, at fixed port 9999. A future release will add a secure port at 9997.

Postman

Postman is an application that you can use to test the Event Master JSON API.

1. Go to the GetPostman website.
   (https://www.getpostman.com/)
2. Download the free Postman app.
3. Install Postman on your machine.
4. Launch Postman and follow these steps to send JSON commands to the Event Master processor.
   a. Select POST from the dropdown next to the URL text box.
   b. Enter HTTP:// followed by the request URL with port 9999.
      1) A future release will add the secure HTTPS:// connection over port 9997.
   c. Type "Content-Type" for header and "application/json" as value of this header.
      (See https://www.getpostman.com/docs/requests for more information on sending requests.)

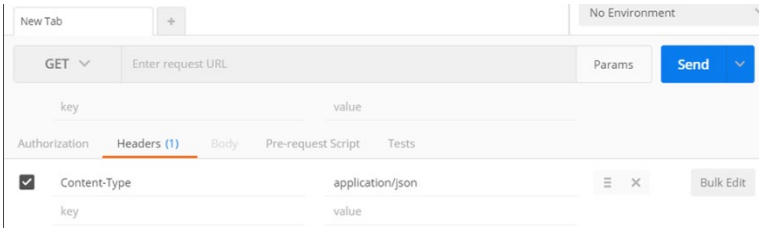

Image B–1
Content-Type and application/json

5. Select ***Body*** and click on raw from the buttons available below the URL text box.
6. Write the request in the body.

A request is a call to a specific method provided by a remote system. It must contain three certain properties:

- method—a String with the name of the method to be invoked.
- params—an Object or Array of values to be passed as parameters to the defined method.
- id—a value of any type used to match the response with the request to which it is replying.

The receiver of the request must reply with a valid response to all received requests. A response must contain the properties mentioned below.

- result—the data returned by the invoked method. If an error occurred while invoking the method, this value must be null.
- error—a specified error code if there was an error invoking the method, otherwise null.
- id—the id of the request to which it is responding.

For example:

```
{"params":{}, "method":"getFrameSettings", "id":"1234", "jsonrpc":"2.0"}
```

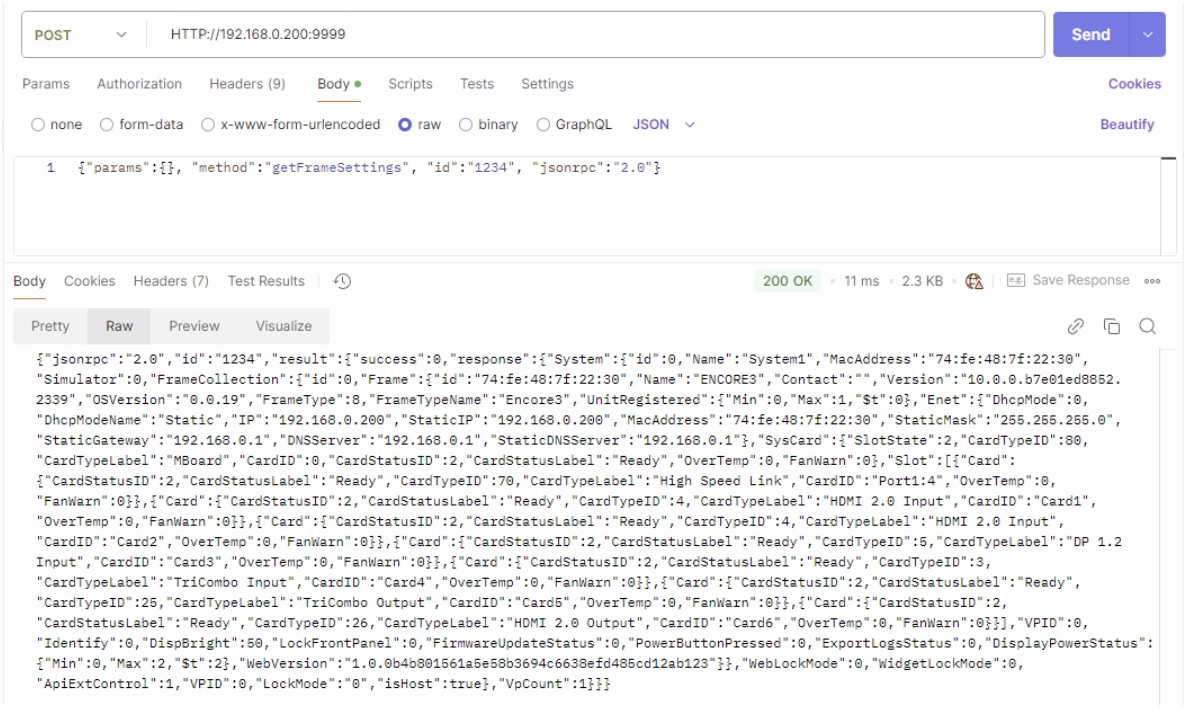

Image B–2
getFrameSettings API – example [powerStatus_API.png]

Here, ***getFrameSettings*** is the JSON API to be sent to an Encore3 video processor.

JSON APIs

Here are some of the JSON APIs defined for controlling the Event Master processor. If you have a host + client(s) Event Master system, JSON must be sent to the host unit. For all the requests, this section explains the parameter passed or used in the RPC calls. If the "params" object is blank that means the API doesn't require a parameter.

Requests are case sensitive.

getFrameSettings

- Definition:
  - This API shows system information, including all the frames information.
- Request:
  - params: {}
- Response:
  - {"jsonrpc":"2.0","id":"1234","result":{"success":0,"response":{"System":{"id":0,"Name":"System1","MacAddress":"74:fe:48:7f:22:30","Simulator":0,"FrameCollection":{"id":0,"Frame":{"id":"74:fe:48:7f:22:30","Name":"ENCORE3","Contact":"","Version":"10.0.0.b7e01ed8852.3640","OSVersion":"0.0.23","FrameType":8,"FrameTypeName":"Encore3","UnitRegistered":{"Min":0,"Max":1,"$t":0},"Enet":{"DhcpMode":0,"DhcpModeName":"Static","IP":"192.168.0.200","StaticIP":"192.168.0.200","MacAddress":"74:fe:48:7f:22:30","StaticMask":"255.255.255.0","StaticGateway":"192.168.0.1","DNSServer":"192.168.0.1","StaticDNSServer":"192.168.0.1"},"SysCard":{"SlotState":2,"CardTypeID":80,"CardTypeLabel":"MBoard","CardID":0,"CardStatusID":2,"CardStatusLabel":"Ready","OverTemp":0,"FanWarn":0},"Slot":[{"Card":{"CardStatusID":2,"CardStatusLabel":"Ready","CardTypeID":70,"CardTypeLabel":"High Speed Link","CardID":"Port1:4","OverTemp":0,"FanWarn":0}},
{"Card":{"CardStatusID":2,"CardStatusLabel":"Ready","CardTypeID":4,"CardTypeLabel":"HDMI 2.0 Input","CardID":"Card1","OverTemp":0,"FanWarn":0}},
{"Card":{"CardStatusID":2,"CardStatusLabel":"Ready","CardTypeID":4,"CardTypeLabel":"HDMI 2.0 Input","CardID":"Card2","OverTemp":0,"FanWarn":0}},
{"Card":{"CardStatusID":2,"CardStatusLabel":"Ready","CardTypeID":5,"CardTypeLabel":"DP 1.2 Input","CardID":"Card3","OverTemp":0,"FanWarn":0}},
{"Card":{"CardStatusID":2,"CardStatusLabel":"Ready","CardTypeID":3,"CardTypeLabel":"TriCombo Input","CardID":"Card4","OverTemp":0,"FanWarn":0}},
{"Card":{"CardStatusID":2,"CardStatusLabel":"Ready","CardTypeID":25,"CardTypeLabel":"TriCombo Output","CardID":"Card5","OverTemp":0,"FanWarn":0}},
{"Card":{"CardStatusID":2,"CardStatusLabel":"Ready","CardTypeID":26,"CardTypeLabel":"HDMI 2.0 Output","CardID":"Card6","OverTemp":0,"FanWarn":0}}],
"VPID":0,"Identify":0,"DispBright":50,"LockFrontPanel":0,"FirmwareUpdateStatus":0,
"PowerButtonPressed":0,"ExportLogsStatus":0,"DisplayPowerStatus":{"Min":0,"Max":2,"$t":2},"WebVersion":"1.0.0b4b801561a5e58b3694c6638efd485cd12ab123"}},"WebLockMode":0,"WidgetLockMode":0,"ApiExtControl":1,"VPID":0,"LockMode":"0","isHost":true},"VpCount":1}}}
    - System — System name and index.
    - FrameCollection — Collection of frames in a system containing frame information.
    - Frame — Contains frame information.
    - Id — Mac Id of the frame.
    - Name — Name of the frame.
    - Contact — Contact information.
    - Version — Current version of the software installed on the frame.
    - OSVersion — Current OS version installed on the frame.
    - FrameType — 0:E2, 1:S3, 2:Ex, 3:IP-4K, 5:E2-Gen2, 6:PDS-4K, 8:Encore3
    - FrameTypeName — Type of the frame: Encore3, E2, S3, Ex, etc.
    - Enet — Ethernet settings.
    - SysCard — System card information.
    - Slot — List of Input/Output/Expansion card information.
  - success: (0=success, anything else is an error)
- Example:
  - {"params":{}, "method":"getFrameSettings", "id":"1234", "jsonrpc":"2.0"}

powerStatus

- Definition:
  - This queries the power plug status of the Event Master processor. (There can be 1 or 2 power slots in Event Master processor).
- Request:
  - params: `{}` - No parameters required.
- Response:
  - response: `{FrameId0:{PwrStatus1, PwrStatus2},{FrameId1 :{ PwrStatus1, PwrStatus2}`
    - PwrStatus1 gives the power status of the 1st slot in Event Master processor with frame id FrameId1, FrameId2.
    - PwrStatus2 gives the power status of the 2nd slot in Event Master processor with frame id FrameId1, FrameId2.
    - 0: Power supply module is not present.
    - 1: Power supply module is present, but there is no AC current detected.
    - 2: Power supply module is present, and the AC current is detected, but there is no DC current.
    - 3: Power supply module is present, and everything is OK.
  - success: (0=success, anything else is an error)
- Example:
  - `{"params":{}, "method":"powerStatus", "id":"1234", "jsonrpc":"2.0"}`

listInputs

- Definition:
  - This queries the list of inputs configured.
- Request:
  - params: `{"inputId": x},`

    "x" can be:
    - −1: For All inputs. (inputId is optional parameter, if not provided list of all inputs configured will be returned.)
    - Anything else will be treated as input config index and will return input of that index.
- Response:
  - response: Array of: `[{"id":8,"Name":"DPInput1","SyncStatus":"None","VideoStatus":"No Sync", "Format":"3840x2160p @59.94", "Color_Space":"RGB, Full Range", "Colorimetry":"BT.709", "GammaFx":"SDR", "ColorDepth":"8"}]`
    - Response contains the array of inputs. Above response contains id, name, Sync status, video status, Format, Color Space, Colorimetry, Gamma Fx, and Color Depth.
  - success: (0=success, anything else is an error)
- Example:
  - `{"params":{"inputId": 1}, "method":"listInputs", "id":"1234", "jsonrpc":"2.0"}`

listOutputs

- Definition:
  - This queries the list of outputs configured.
- Request:
  - params: `{"outputCfgId": x},`

    "x" can be:
    - −1: For All outputs. (outputCfgId is not mandatory parameter, if not provided list of all outputs configured will be returned.)
    - Anything else will be treated as output config index and will return output of that index.
- Response:
  - response: Array of: `[{"id":2,"Name":"HDMIOutput3","Format":"3840x2160p CEA @59.94","ColorSampleBit":"RGB/4:4:4/8","Color_Space":"RGB, Full Range","Colorimetry":"BT.709","GammaFx":"SDR","HDRMetaFileIndex":0}]`
    - Response contains the array of outputs. Above response contains id, name, Format, Color/Sample/Bit, Color space, Colorimetry, Gamma Fx, and HDRMetaFileIndex.
  - success: (0=success, anything else is an error)
- Example:
  - `{"params":{"outputCfgId": 1}, "method":"listOutputs", "id":"1234", "jsonrpc":"2.0"}`

resetFrameSettings

- Definition
  - Send a reset type to an Encore3 processor with different options.
- Request:
  - params: `{"reset":x},`

    "x" can be 0 – 3
    - 0: Soft Reset.
    - 1: Factory Reset.
    - 2: Factory Reset and Keep.

      - add params: `"saveOptions":y`

      "y" can be 0 – 4 (0 is optional)
      - 0: All
      - 1: IP Settings
      - 2: Input EDIDs
      - 3: Unit ID
      - 4: System Native Rate
    - 3: Power Down.
- Response:
  - response: null
  - success: (0=success, anything else is an error)
- Example
  - `{"params":{"reset":0}, "method":"resetFrameSettings", "id":"1234", "jsonrpc":"2.0"}`
    // Sends the Soft Reset command
  - `{"params":{"reset":2, "saveOptions":4}, "method":"resetFrameSettings", "id":"1234", "jsonrpc":"2.0"}`
    // Sends the Factory and Keep System Native Rate Reset command

listOperators

- Definition:
  - This queries the list of operators configured.
- Request:
  - params: `{}`
- Response:
  - response: Array of `[{"id":0,"Name":"Operator 1", "Enable":0, "StartRange":1, "EndRange":1000, "InvertColor":0,"DestCollection":[]},{"id":1,"Name":"Operator 2", "Enable":1, "StartRange":3, "EndRange":4, "InvertColor":0, "DestCollection":[{"id":0, "DestType":1, "DestXmlId":0, "Name":"ScreenDest1"}, {"id":1, "DestType":0, "DestXmlId":0, "Name":"AuxDest1"}]}, {"id":2,"Name":"Operator 3", "Enable":0, "StartRange":1, "EndRange":1000, "InvertColor":0, "DestCollection":[{"id":0, "DestType":1, "DestXmlId":0, "Name":"ScreenDest1"}]}]`
    - Response contains the array of multi-operators. Above response contains id, name, Enable mode, Start Range of preset serial number, End Range of preset serial number, color of controller is inverted for this operator and collection of destination selected for this operator.
  - success: (0=success, anything else is an error)
- Example:
  - `{"params":{}, "method":"listOperators", "id":"1234", "jsonrpc":"2.0"}`

configureOperator

- Definition:
  - This API helps user to configure operators.
- Request:
  - params: `{"operatorId": 2, "name": "qwert", "startRange":89, "endRange":95, "enable": 0, "add":{"destType": 1, "destIndex":0}, "remove" :{"destType": 1, "destIndex":1}}`
    - `operatorId`: Operator index which needs to be configured.
    - `name`: User can set the name of the operator.
    - `startRange` : This is start range of preset serial number assigned to the operator.
    - `endRange`: This is end range of preset serial number assigned to the operator.
    - `enable`: enable (1) or disable (0) the operator.
    - `add/remove`: Add/Remove destination for the operator.
      - destType: 0 – aux, 1 – screen.
      - destIndex: destination index.
- Response:
  - response: null
  - success: (0=success, anything else is an error)
- Examples:
  - `{"params": {"operatorId": 2, "name": "qwert", "startRange":89, "endRange":95, "enable": 1}, "method":"configureOperator", "id":"1234", "jsonrpc":"2.0"}`
    // Enables Operator 3 while changing its name to qwert and assigns preset range 89-95.
  - `{"params": {"operatorId": 1, "add":{"destType": 1, "destIndex":0}, "remove": {"destType": 1, "destIndex":1}}, "method":"configureOperator", "id":"1234", "jsonrpc":"2.0"}`
    // For Operator 2 – Removes Screen Destination 2 and adds Screen Destination 1 for this operator to control.
  - `{"params": {"operatorId": 2, "name": "operator3", "endRange":134}, "method":"configureOperator", "id":"1234", "jsonrpc":"2.0"}`
    // For Operator 3 – Renames the profile to operator3 and changes the ending preset in its range to 134.
  - `{"params": {"operatorId": 1, "name": "operator2", "startRange":21}, "method":"configureOperator", "id":"1234", "jsonrpc":"2.0"}`
    // For Operator 2 – Renames the profile to operator2 and changes the starting preset in its range to 21.

listSources

- Definition:
  - This API lists all the input sources with properties.
- Request:
  - params: `{}`
- Response:
  - response: Array of: `{"id": 0, "Name": "InSource1", "HSize": 3840, "VSize": 1080, "SrcType": 0, "InputCfgIndex": -1, "StillIndex": 0, "DestIndex": -1, "UserKeyIndex": -1, "Mode3D": 0, "Freeze": 1, "Capacity": 2, "InputCfgVideoStatus": 4}`
    success: (0=success, anything else is an error)
  - Parameter to look for is "InputCfgVideoStatus". Possible values:
    0 = Invalid; there is sync, but cannot acquire / lock mismatch
    1 = Valid; Video is OK
    2 = MismatchFormat; Format mismatch between input cfg and connector(s)
    3 = OutOfRange; connector capacity is too low to acquire format
    4 = NoSync; no video
- Example:
  - `{"params": {}, "method":"listSources", "id":"1234", "jsonrpc":"2.0"}`

listStill

- Definition:
  - This API lists all the stills with properties such as id, Name, H/V size, LockMode, StillState, PngState, File size.
- Request:
  - params: `{}`
- Response:
  - response: Array of :
    `[{"id":0,"Name":"a.png","LockMode":{"Min":0,"Max":1,"$t":0},"HSize":{"Min":0,"Max":99999,"$t":3840},"VSize":{"Min":0,"Max":99999,"$t":2160},"AlphaMode":{"Min":0,"Max":1,"$t":0},"UmdColorStatus":{"LegalValues":"0 1 2 3 4 5","$t":1}, "StillState":{"Min":0,"Max":4,"$t":3}, "PngState":{"Min":0,"Max":2,"$t":0},"FileSize":{"Min":0,"Max":100000,"$t":33177.6}}]`
    - id — Index of still store.
    - Name — Name of still store.
    - LockMode — For future use. Always set to 0.
    - H/V size — Horizontal and vertical size, Min, max and current value. It shows the current value.
    - StillState — This tells user if the still is currently being captured or not, or if it is getting deleted.
    - PngState — The "PNG" for stills are for the thumbnails we capture for the stills.
    - FileSize — Size of the file created in KBs.
  - success: (0=success, anything else is an error)
- Example:
  - `{"params": {}, "method":"listStill", "id":"1234", "jsonrpc":"2.0"}`

takeStill

- Definition:
  - This API captures/overwrites a still.
- Request:
  - params: `{"type": 0, "id": y, "file": z}`
    - type — 0 for input source
    - Id — Index of the input. If the input id of the destination is provided, no still is created and an error is shown.
    - File — still file id. If you pass "file" : 5, this creates StillStore6.
- Response:
  - response: null
  - success: (0=success, anything else is an error)
- Example:
  - `{"params":{"type":0, "id": 1, "file": 5}, "method":"takeStill", "id":"1234", "jsonrpc":"2.0"}`
    // This captures a still from input id 1 and saves it as StillStore6.

deleteStill

- Definition:
  - This API deletes a still.
- Request:
  - params: `{"id": x}`
    - id—Index of still.
- Response:
  - response: null
  - success: (0=success, anything else is an error)
- Example:
  - `{"params": {"id": 0}, "method":"deleteStill", "id":"1234", "jsonrpc":"2.0"}`

listSourceMainBackup

- Definition:
  - This API shows a list of inputs which have at least one backup input or still configured.
- Request:
  - params: `{"inputType": x}`
    - x can be – 1 (default) for all Inputs, or 0-99 for individual input ID. This is not a mandatory parameter. If not provided, all are listed.
- Response:
  - response: Array of inputs / Background
    `[{"id":0,"Name":"HDMIInput1","Backup":[{"id":0,"inputId":8,"stillId":null,"destId":null,"Name":"DPBackground1"},{"id":2,"inputId":4,"stillId":null,"destId":null,"Name":"DPInput5"}]}]`
    - id — Index of input ID.
    - Name — Name of input.
    - VideoStatus:
      - 0: there is sync, but cannot acquire / lock mismatch
      - 1: Video status is OK
      - 2: Format mismatch between Input config and connector(s)
      - 3: Capacity / system mode error
      - 4: A connector lost sync
    - Array of Backup
      - Id — backup Index (0, 1, 2). (Max we can set 3 backup)
      - InputId: index of input which is configured as backup source.
      - stillId: index of still which is configured as backup source.
      - destId: index of screen destination which is configured as backup source.
      - Name: Name of source configured as backup.
  - success: (0=success, anything else is an error)
- Example:
  - `{"params":{"inputType": -1}, "method":"listSourceMainBackup", "id":"1234", "jsonrpc":"2.0"}`

activateSourceMainBackup

- Definition:
  - This API configure backups on inputs and backgrounds.
- Request:
  - params: `{"inputId":8,`
    `"Backup1": {"SrcType": 1, "SourceId": 1},`
    `"Backup2": {"SrcType": 0, "SourceId": 0},`
    `"Backup3": {"SrcType": 1, "SourceId": 0},`
    `"BackUpState":1}`
    - inputId: index of input/Background for which backup needs to be configured.
    - Backup1/Backup2/Backup3:
      - SrcType: 0 for input, 1 for Stills.
      - SourceId: Index of input or Still.
    - BackupState: Backup id which needs to be set for backup of the main input. -1 to set primary and is default (If not provided then primary will be activated)
- Response:
  - response: null
  - success: (0=success, anything else is an error)
- Example:
  - `{"params":{"inputId":8,`
    `"Backup1": {"SrcType": 1, "SourceId": 1},`
    `"Backup2": {"SrcType": 0, "SourceId": 0},`
    `"Backup3": {"SrcType": 1, "SourceId": 0},`
    `"BackUpState":1}, "method":"activateSourceMainBackup", "id":"1234",`
    `"jsonrpc":"2.0"}`

resetSourceMainBackup

- Definition:
  - This API resets the applied Input backup to primary source.
- Request params:
  - params: `{"id": 2}`
    - id: Input index ID to be reset.
- Response:
  - response: null
  - success: (0=success, anything else is an error)
- Example:
  - `{"params": {"id": 22}, "method":"resetSourceMainBackup", "id":"1234",`
    `"jsonrpc":"2.0"}`

listDestinations

- Definition:
  - This API lists all the destinations with properties such as layers, outputs, ID, size, and name.
- Request:
  - params: `{"type": x}`
    - 0 — Show all the destinations.
      0 is the default value for the parameter.
    - 1 — Only screen destinations.
    - 2 — Only aux destinations.
- Response:
  - response: Array of : `{"ScreenDestination":[{"id": 0, "Name": "Dest1", "HSize": 7680,`
    `"VSize": 2160, "Layers": null,"DestOutMapColl":[{"id": 0"DestOutMap":[{"id": 0,`
    `"Name": "HDMIOutput1", "HPos": 0, "VPos": 0, "HSize": 3840, "VSize":2160, "Freeze":`
    `0},{"id": 1, "Name": "Out2", "HPos": 3840, "VPos": 0, "HSize": 3840, "VSize":2160,`
    `"Freeze": 0}]}]}],"AuxDestination":[{"id": 0, "AuxStreamMode": 4}, {"id": 1,`
    `"AuxStreamMode": 4}]}"`
  - success: (0=success, anything else is an error)
- • Example:
  - `{"params": {"type": 0}, "method":"listDestinations", "id":"1234", "jsonrpc":"2.0"}`

listAuxContent

- Definition:
  - This API shows Aux destination information.
- Request:
  - params: `{"id": x}`
    - Id — Index of the Aux destination.
- Response:
  - response: `{"id":0,"Name":"AuxDest1","PvwLastSrcIndex":0,"PgmLastSrcIndex":0}`
    - id — Index of Aux destination.
    - Name — Name of Aux destination.
    - PvwLastSrcIndex — Input source index in the preview area.
    - PgmLastSrcIndex — Input source index in the program area.
  - success: (0=success, anything else is an error)
- Example:
  - `{"params": {"id": 0}, "method":"listAuxContent", "id":"1234", "jsonrpc":"2.0"}`

changeAuxContent

- Definition:
  - This API changes the source in the Aux destinations.
- Request:
  - params: `{"id":x, "Name": "AuxDest1", "PvwLastSrcIndex": y, "PgmLastSrcIndex": z}`
    - id — Index of the Aux destination.
    - Name — Name of Aux destination. (Optional paramter)
    - PvwLastSrcIndex — Input source index ID to set in Aux destination in the preview area.
    - PgmLastSrcIndex — Input source index ID to set in Aux destination in the program area.
    - TestPattern — Provide test pattern index ID number
- Response:
  - response: null
  - success: (0=success, anything else is an error)
- Examples:
  - `{"params": {"id":0, "Name": "AuxDest1", "PvwLastSrcIndex": 6, "PgmLastSrcIndex": 1}, "method":"changeAuxContent", "id":"1234", "jsonrpc":"2.0"}`
    // Places Source ID 6 into preview and Source ID 1 on program for Aux Destination 1.
  - `{"params":{"id":0, "TestPattern" :3 }, "method":"changeAuxContent", "id":"1234", "jsonrpc":"2.0"}`
    // Turns on the 100% Color Bars Test Pattern in Aux Destination 1.

listContent

- Definition:
  - This API shows the content of a screen destination.
- Request:
  - params: `{"id": x, "DestOutMap": y}`
    - "id" — Screen destination index.
    - "DestOutMap" – Destination's Output index number (Optional)
- Response:
  - response:`{"jsonrpc":"2.0","result":{"success":0,"response":{"id":0,"Name":"ScreenDest1","IsActive":1,"BGLyr":[{"id":0,"LastBGSourceIndex":-1, "BGShowMatte":1, "BGColor":{"id":0, "Red":0, "Green":0, "Blue":0}},{"id":1, "LastBGSourceIndex":-1, "BGShowMatte":1, "BGColor":{"id":0, "Red":0, "Green":0, "Blue":0}}], "Layers":[{"id":0, "Name":"Layer1-A","LastSrcIdx":-1, "PvwMode":0, "PgmMode":0, "LinkLayerId":0, "LinkDestId":0, "Capacity":1, "PvwZOrder":0, "PgmZOrder":0, "Freeze":0, "ScalingMode":2, "Window":[{"HPos":0, "VPos":0, "HSize":1920, "VSize":1080}, {"HPos":0, "VPos":0, "HSize":1920, "VSize":1080}], "Source":[{"HPos":0,"VPos":0,"HSize":1920,"VSize":1080},{"HPos":0,"VPos":0,"HSize":1920,"VSize":1080}],"Mask":[{"id":0,"Top":0,"Left":0,"Right":0,"Bottom":0},{"id":0,"Top":0,"Left":0,"Right":0,"Bottom":0}]},{"id":1,"Name":"Layer1-B", "LastSrcIdx":-1, "PvwMode":0, "PgmMode":0, "LinkLayerId":1, "LinkDestId":0, "Capacity":1, "PvwZOrder":0, "PgmZOrder":0, "Freeze":0, "ScalingMode":2, "Window":[{"HPos":0,"VPos":0,"HSize":1920,"VSize":1080},{"HPos":0,"VPos":0,"HSize":1920,"VSize":1080}],"Source":[{"HPos":0,"VPos":0,"HSize":1920,"VSize":1080},{"HPos":0,"VPos":0,"HSize":1920,"VSize":1080}],"Mask":[{"id":0,"Top":0,"Left":0,"Right":0,"Bottom":0},{"id":0,"Top":0,"Left":0,"Right":0,"Bottom":0}]}],"Transition":[{"id":0,"Tr`

ansTime":30,"TransPos":0,"ArmMode":1},{"id":1,"TransTime":30,"TransPos":0,"ArmMode":0}],"OutputCfg":[{"id":0,"Name":"HDMIOutput1","OutputAOI":[{"id":0,"TestPattern":[{"id":0,"TestPatternMode":0}]}]}]}},"id":"1234"}

  - id — index of screen destination.
  - Name — Name of Screen Destination.
  - BGLyr — Background layer index, Last source index of background. "id":0 affects the Background in Program. "id":1 affects the Background in Preview.
  - LastBGSoureIndex — This is –1 if no background is dropped (Matte Color), else this is the index of the last input source used on BG of the screen destination.
  - BGShowMatte — This is if BG to be matte or not.
  - BGColor — This is the background color.
  - Layers — Lists layers on screen destination with its properties.
  - Transition — This property of screen destination contains the transition time (from time to move from preview to program).
  - LinkLayerId: Link/Global Layer Index
  - LinkDestId: Link Destination Index

- success: (0=success, anything else is an error)

- Example:
  - `{"params": {"id": 0}, "method":"listContent", "id":"1234", "jsonrpc":"2.0"}`
    // Lists the BG and layers contents of Screen 1
  - `{"params": {"id": 1,"DestOutMap":2}, "method":"listContent", "id":"1234", "jsonrpc":"2.0"}`
  - *// Lists Screen 2's BG sources and contents of the 3rd Output's layers.*

changeContent

- Definition:
  - This API changes the content of a screen destination by putting background and layers in it.
- Request:
  - params: `{"id":0, "DestOutMap":0, "TestPattern" :5, "BGLyr":[{"id":0,"LastBGSourceIndex":0,"BGShowMatte":0, "BGColor":[{"id":0,"Red":0,"Green":0,"Blue":0}]},{"id":1,"LastBGSourceIndex":0, "BGShowMatte":0,"BGColor":[{"id":0,"Red":0,"Green":0,"Blue":0}]}],"Layers": [{"id":0,"LastSrcIdx":0,"Window":{"HPos":0,"VPos":0,"HSize":400,"VSize":300}, "Source":{"HPos":0,"VPos":0,"HSize":1920,"VSize":1080}, "Mask":{ "Left":0.01, "Right":10.1, "Top":0.0,"Bottom":0.0},"PvwMode":1,"PgmMode":0}]}`

    - id — Screen destination index.
    - DestOutMap – Index of Screen Destination's output (Optional)
    - BGLyr — Background layer index, Last source index of background. "id":0 affects the Background in Program. "id":1 affects the Background in Preview.
    - Layers — Layer information collection.
      - id — Layer index number.
      - LastSrcIdx — Source Index number.
      - Window — Layer window size.
      - Source — Source size and offset.
      - Mask — Crop the visible part of the layer. Mask parameters must be sent separately from Window and LastSrcIdx.
    - PvwMode — Set 1 if you want the content in preview. (Default)
    - PgmMode — Set 1 if you want the content in program.
    - TestPattern — Provide test pattern id (0=Off).
- Response:
  - response: null
  - success: (0=success, anything else is an error)

- Examples:
  - `{"params":{"id":0,"BGLyr":[{"id":0,"LastBGSourceIndex":-1, "BGShowMatte":1,"BGColor":[{"id":0,"Red":0,"Green":0,"Blue":0}]},{"id":1,"LastBGSourceIndex":-1,"BGShowMatte":1,"BGColor":[{"id":0,"Red":0,"Green":1023,"Blue":0}]}]}, "method":"changeContent","id":"1234","jsonrpc":"2.0"}`
    // Sets the Backgrounds for Destination 1 to use a black matte on PGM and full green matte on PVW.
  - `{"params": {"id": 1, "DestOutMap": 1, "Layers": [{"id": 4, "LastSrcIdx":7, "Window":{"HPos": 1540, "VPos": 680, "HSize": 1600, "VSize": 900}, "PvwMode":1}]}, "method":"changeContent", "id":"1234", "jsonrpc":"2.0"}`
    // Places Source ID 7 into Layer 3.2A on Destination 2's PVW at the given window size and position.
  - `{"params": {"id": 1, "DestOutMap": 0, "Layers": [{"id": 4, "Mask":{"Left":12.5, "Right":12.5}, "PvwMode":1}]}, "method":"changeContent", "id":"1234", "jsonrpc":"2.0"}`
    // Sets the Mask values for Layer 3.1A on Destination 2's PVW to 12.5% on the left and right edges.
  - `{"params": {"id":1, "DestOutMap": 2, "TestPattern":5}, "method":"changeContent", "id":"1234", "jsonrpc":"2.0"}`
    // Turns on the 32x32 Grid Test Pattern for Output 3 of Screen Destination 2.

listUserKey

- Definition:
  - This API lists all User Keys in the system.
- Request params:
  - params: `{}`
- Response:
  - response: null
  - success: (0=success, anything else is an error)
- Example:
  - `{"params": {}, "method":"listUserKeys", "id":"1234", "jsonrpc":"2.0"}`

recallUserKey

- Definition:
  - Recall a User Key on the Encore3 processor. Users can recall User Key with id or User Key name. Send any one of the parameters to recall User Key.
- Request params:
  - params: `{"id": x, "ScreenDestination": [], "DestOutMap": [], "Layer":[] }`
  - params: `{"userkeyName":"UserKeyName", "ScreenDestination": [], "DestOutMap": [],"Layer":[] }`
    - id — Index of User Key
    - userKeyName — Name of a User Key
    - ScreenDestination — Array using indices of screen destinations.
    - DestOutMap — Array using indices of Destination Output Maps containing output layers.
    - Layer — Array of layer indices in the screen destination on which the User Key is to be recalled.
      ***Note**: If user provides params: {"id": 3, "ScreenDestination": [1,2], "DestOutMap": [0,1], "Layer":[1,2,3]} this means apply user key number 4 to:*
      Screen 2 – 1st Output – Layers 2,3,4
      Screen 2 – 2nd Output – Layers 2,3,4
      Screen 3 – 1st Output – Layers 2,3,4
      Screen 3 – 2nd Output – Layers 2,3,4
- Response:
  - response: null
  - success: (0=success, anything else is an error)
- Examples:
  - `{"params": {"id": 0, "ScreenDestination": [0,1,2], "Layer":[0,2,4]}, "method":"recallUserKey", "id":"1234", "jsonrpc":"2.0"}`
    // Applies UserKey ID 0 to Layers 1, 3 & 5 on Screens 1, 2 & 3.
  - `{"params": {"userkeyName": "abc", "ScreenDestination": [0,1], "DestOutMap": [0,3], "Layer":[0]}, "method":"recallUserKey", "id":"1234", "jsonrpc":"2.0"}`
    // Applies UserKey name "abc" to Layer 1 to Output layers 1.1A & 1.4A in Screens 1 & 2.

fillHV

- Definition:
  - Fits layers to screen destination horizontally and vertically.
- Request:
  - params: `{"screenId": x, "DestOutMap": y, "Layers": [{"id": 0}, {"id": 1}, {"id": 3}]}`
    - screenId — index ID of screen destination
    - DestOutMap — index ID of destination output
    - Layers — List of layer ID indices.
- Response:
  - response: null
  - success: (0=success, anything else is an error)
- Example:
  - `{"params": {"screenId": 0, "Layers": [{"id": 0}, {"id": 1}]}, "method":"fillHV", "id":"1234", "jsonrpc":"2.0"}`
    // Layers 1A & 1B in Screen 1 will fill the screen's entire width and height.
  - `{"params": {"screenId": 0, "DestOutMap": 1, "Layers": [{"id": 2}]}, "method":"fillHV", "id":"1234", "jsonrpc":"2.0"}`
    // Output Layer 2.2A in Screen 1 will fill the screen's entire width and height.

clearLayers

- Definition:
  - Clear layers from screen destinations.
- Request:
  - params: `{"screenId": x, "DestOutMap": y, "Layers": [{"id": 0}, {"id": 1}, {"id": 2}]}`
    - screenId — index ID of screen destination
    - DestOutMap — index ID of destination output
    - Layers — List of layer ID indices.
- Response:
  - response: null
  - success: (0=success, anything else is an error)
- Example:
  - `{"params": {"screenId": 0, "Layers": [{"id": 0}, {"id": 1}]}, "method":"clearLayers", "id":"1234", "jsonrpc":"2.0"}`
    // Clears Layers 1A & 1B from Screen 1.
  - `{"params": {"screenId": 1, "DestOutMap": 2, "Layers": [{"id": 6}, {"id": 7}]}, "method":"clearLayers", "id":"1234", "jsonrpc":"2.0"}`
    // Clears Output Layers 4.3A & 4.3B from Screen 2.

listPresets

- Definition:
  - This queries the list of Presets on a particular destination or on the system.
- Request:
  - params: `{"ScreenDest":x, "AuxDest":x},`

    "x" can be:
    - −2: Do not include any destinations of this type. (Has priority over particular id, if passed as a parameter.)
    - −1: Do not care (All presets). (Has priority over particular ID, if passed as a parameter.)
    - 0–999: want to see the presets with the destination's particular id in it or array of ids. E.g. "ScreenDestination":[{"id": 2}, {"id": 3}]
- Response:
  - response: Array of: `[{"id": 0, "Name": "Preset3.00", "LockMode": 0, "presetSno": 3.00}, {"id": 1, "Name": "Preset4.00", "LockMode": 0, "presetSno": 4.00}]`
    - Response contains the array of presets. Above response contains id, name, lock mode preset serial number of the all the presets.
  - success: (0=success, anything else is an error)
- Example:
  - `{"params":{"ScreenDest": -1}, "method":"listPresets", "id":"1234", "jsonrpc":"2.0"}`

savePreset

- Definition:
  - Creates a Preset on the Encore3 processor.
- Request:
  - params: `{"presetName": "NewPreset", "ScreenDestination":[{"id": 2}, {"id": 3}], "AuxDestination":[{"id": 1}, {"id": 2}]}`
  - params: `{"presetName": "NewPreset", "serialNo": 1.01, "saveFromProgram":1, "ScreenDestination":[{"id": 2}, {"id": 3}], "AuxDestination":[{"id": 1}, {"id": 2}]}`
    - presetName — Name of the Preset to save.
    - ScreenDestinations — ScreenDest id for the Preset to be created.
    - AuxDestinations — AuxDest id for the Preset to be created.
    - ScreenDestination, AuxDestinations are optional parameters. If user doesn't provide it, Preset will be saved for selected destinations.
    - serialNo — serial number for the preset to be saved. If the serialNo exists, it will be overwritten. (Optional). Only two-digit decimal points are recommended. If user provides more than 2 decimal places the number may be rounded off to two-digit decimal place (hundredths).
    - saveFromProgram — This flag is set to 1 if the preset is to be saved from program, else the default will be saved from preview. (Optional)
- Multi-Operator Mode:
  New parameters are introduced to work with multi-operator mode along with the above parameters.
  These parameters are used only when one or more operators are enabled.
  - params: `{"presetName": "NewPreset", "operatorId": y} (for normal operator)`
    - "operatorId" — operator index (For current release only 0, 1, 2 are valid indices).
    - If user wants to use "super-operator" mode, its password is required which is passed as a parameter.
  - params: `{"presetName": "NewPreset", "password": "xyz"} (for super operator)`
    - password — Super user password saved. When this is passed, actions will be performed as if no operator is enabled.
- Response:
  - response: null
  - success: (0=success, anything else is an error)
- Examples:
  - `{"params": {"presetName": "NewPreset"}, "method":"savePreset", "id":"1234", "jsonrpc":"2.0"}`
    // Saves a preset named "NewPreset" from the selected destinations' previews as the next available preset in the list.
  - `{"params": {"presetName": "NewPreset-S1-A1", "ScreenDestination": {"id": 0},"AuxDestination":{"id": 0}}, "method":"savePreset", "id":"1234", "jsonrpc":"2.0"}`
    // Selects Screen1 and Aux1 only and saves a preset from their previews named "NewPreset-S1-A1" as the next available preset in the list.
  - `{"params": {"presetName": "NewSubPreset", "serialNo": 1.01, "saveFromProgram":1}, "method":"savePreset", "id":"1234", "jsonrpc":"2.0"}`
    // Saves a preset named "NewSubPreset" from the selected destinations' programs as the preset number 1.01 in the list.

  For normal operator:
  - `{"params": {"presetName": "NewPreset5", "serialNo": 5.00, "operatorId": 2}, "method":"savePreset", "id":"1234", "jsonrpc":"2.0"}`
    // Saves a preset named "NewPreset5" from the selected destinations' previews with preset number 5.00 in Operator 3's list.

  For super operator:
  - `{"params": {"presetName": "NewPreset", "password": "p@$$w0rD"}, "method":"savePreset", "id":"1234", "jsonrpc":"2.0"}`
    // Saves a preset named "NewPreset" from the selected destinations' previews as the next available preset in the list.

renamePreset

- Definition:
  - Rename a Preset on the Encore3 system. User can rename Preset with id, Preset serial number, or Preset name. Send any one of the parameters to rename Preset.
- Request params:
  - params: `{"id": x, "newPresetName": "NewPresetName"}`
  - params: `{"presetSno": x.y, "newPresetName": "NewPresetName"}`
  - params: `{"presetName": "OldPresetName", "newPresetName": "NewPresetName"}`
    - "newPresetName" — New Preset name to set.
- Response:
  - response: null
  - success: (0=success, anything else is an error)
- Examples:
  - `{"params": {"id": 0, "newPresetName": " newPresetName "}, "method":"renamePreset", "id":"1234", "jsonrpc":"2.0"}`
  - `{"params": {"presetName": "NewPreset", "newPresetName": "NewPresetName"}, "method":"renamePreset", "id":"1234", "jsonrpc":"2.0"}`
  - `{"params": {"presetSno": 1.00, "newPresetName": " newPresetName "}, "method":"renamePreset", "id":"1234", "jsonrpc":"2.0"}`

deletePreset

- Definition:
  - Delete a Preset on the Encore3 system.
    User can delete Preset with id, Preset serial number, or Preset name.

Send any one of the parameters to delete Preset.

- Request:
  - params: `{"id": x}`
  - params: `{"presetSno": x.y}`
  - params: `{"presetName": "PresetName"}`
- Multi-Operator Mode:
  New parameters are introduced to work with multi-operator mode along with the above parameters.
  These parameters are used only when one or more operators are enabled.
  - params: `{"id": x, "operatorId": y} (for normal operator)`
    - "operatorId" — operator index (For current release only 0, 1, 2 are valid indices).
    - If user wants to use "super-operator" mode, its password is required which is passed as a parameter.
  - params: `{"id": x, "password": "xyz"} (for super operator)`
    - password — Super user password saved. When this is passed, actions will be performed as if no operator is enabled.
- Response:
  - response: null
  - success: (0=success, anything else is an error)
- Examples:
  - `{"params": {"id": 1}, "method":"deletePreset", "id":"1234", "jsonrpc":"2.0"}`
  - `{"params": {"presetSno": 1.00}, "method":"deletePreset", "id":"1234", "jsonrpc":"2.0"}`
  - `{"params": {"presetName": "Preset 5.00"}, "method":"deletePreset", "id":"1234", "jsonrpc":"2.0"}`

  For super operator:
  - `{"params": {"id": 6, "password": "123"}, "method":"deletePreset", "id":"1234", "jsonrpc":"2.0"}`

  For normal operator:
  - `{"params": {"id": 5, "operatorId": 2}, "method":"deletePreset", "id":"1234", "jsonrpc":"2.0"}`

activatePreset

- Definition:
  - Recall a Preset on the Encore3 system. User can recall Preset with id, Preset serial number, or Preset name. Send any one of the parameters to recall Preset.
- Request params:
  - params: `{"id": x, "type": y}`
  - params: `{"presetSno": x.y, "type": y}`
  - params: `{"presetName": "PresetName"}`
    - "type" — 0 to recall in preview (default), 1 to recall in program. This is not a mandatory parameter but should be given when the user wants to recall a Preset to program.
- Multi-Operator Mode:
  - New parameters are introduced to work with multi-operator mode along with the above parameters. These parameters are used only when one or more operators are enabled.
  - params: `{"id": x, "operatorId": y} (for normal operator)`
    - "operatorId" — operator index (Only 0, 1, 2 are valid index IDs).
    - If user wants to use "super-operator" mode, its password is required which is passed as a parameter.
  - params: `{"id": x, "password": "xyz"} (for super operator)`
    - password — Super user password saved. When this is passed, actions will be performed as if no operator is enabled.
- Response:
  - response: null
  - success: (0=success, anything else is an error)
- Examples:
  - `{"params": {"id": 0, "type": 0}, "method":"activatePreset", "id":"1234", "jsonrpc":"2.0"}` *//Recall preset id 0 to preview.*
  - `{"params": {"presetName": "abc"}, "method":"activatePreset", "id":"1234", "jsonrpc":"2.0"}` *//Recall preset named "abc" to preview.*
  - `{"params": {"presetSno": 1.30, "type": 1}, "method":"activatePreset", "id":"1234", "jsonrpc":"2.0"}` *//Recall preset serial number 1.30 to program.*

  For super operator:
  - `{"params": {"id": 6, "password": "123"}, "method":"activatePreset", "id":"1234", "jsonrpc":"2.0"}`

  For normal operator:
  - `{"params": {"id": 5, "operatorId": 2}, "method":"activatePreset", "id":"1234", "jsonrpc":"2.0"}`

recallNextPreset

- Definition:
  - Recall the next Preset by Index ID on the Encore3 system.
  - No parameter is required.
  - Make sure that the user has recalled at least one Preset. Web app recalls the next Preset Index ID following the previously recalled Preset ID.
- Request:
  - params: `{}`
- Response:
  - response: null
  - success: (0=success, anything else is an error)
    An error is shown if there was not a previously recalled Preset or if there is no next Preset in the list.
- Example:
  - `{"params": {}, "method":"recallNextPreset", "id":"1234", "jsonrpc":"2.0"}`

listDestinationsForPreset

- Definition:
  - Lists the Destination content of a Preset.
- Request:
  - params: `{"id":x},`

    "x" can be:
    - −1: List all Presets.
    - 0–999: list only that specific Preset.
- Response:
  - response: Array of: `[{"id": 0, "Name": "Preset3.00", "LockMode": 0, "presetSno": 3.00, "ScreenDest":[{"id": 0}, {"id": 3}],"AuxDest":[{"id": 0}, {"id": 1}]}]`
    - Response contains the array of Presets.
  - success: (0=success, anything else is an error)
- Example:
  - `{"params":{"id": 0}, "method":"listDestinationsForPreset", "id":"1234", "jsonrpc":"2.0"}`

freezeDestSource

- Definition:
  - This API Freezes/Unfreezes the sources.
- Request:
  - params: `{"type": x, "id": y, "screengroup": 0, "mode": 0/1}`
    - type — type of source.
      - 0 — Input.
      - 2 — ScreenDestination.
      - 3 — AuxDestination.
    - id — Index of the Type.
    - Screengroup — For future use. Always set to 0.
    - Mode — 0: UnFreeze, 1: Freeze.
- Response:
  - response: null
  - success: (0=success, anything else is an error)
- Example:
  - `{"params": {"type":0, "id":0, "screengroup":0, "mode":1}, "method":"freezeDestSource", "id":"1234", "jsonrpc":"2.0"}`

armUnarmDestination

- Definition:
  Arm and Unarm the destinations for transitions and saving presets.
- Request:
  - params: `{"arm": x, "ScreenDestination":[{"id": 2}, {"id": 3}], "AuxDestination":[{"id": 1}, {"id": 2}]}`
    - "arm": "x" can be: 0 to unarm and 1 to arm.
    - ScreenDestinations — ScreenDest IDs to arm/unarm.
    - AuxDestinations — AuxDest IDs to arm/unarm.
- Response:
  - response: null
  - success: (0=success, anything else is an error)
- Example:
  - `{"params": {"arm": 1, "ScreenDestination":[{"id": 0}, {"id": 2}], "AuxDestination":[{"id": 0}, {"id": 1}]}, "method":"armUnarmDestination", "id":"1234", "jsonrpc":"2.0"}`

activateDestGroup

- Definition

- Recall a DestGroup on the Encore3 system. User can recall DestGroup with id, DestGroup serial number, or DestGroup name.
- Send any one of the parameters to recall DestGroup.

- Request params:
  - params: `{"id": x}`
  - params: `{"destGrName": "GroupName"}`
    - id – Index of the Destination group.
    - destGrName – Destnation group name.
- Response:
  - Response: null
  - success: (0=success, anything else is an error)
- Example:
  - `{"params": {"id": 0}, "method":"activateDestGroup", "id":"1234", "jsonrpc":"2.0"}`
  - `{"params": {"destGrpName": "abc"}, "method":"activateDestGroup", "id":"1234", "jsonrpc":"2.0"}`

allTrans

- Definition
  - It executes the "allTrans" command.
  - If multi-operator mode is enabled, all-trans will affect only on those destinations which are selected for the requested operator.
- Request:
  - param: {"transTime": x} (optional) — integer value for the number of frames to use for the transition time. This will be applied to all armed destinations.
- Multi-Operator Mode:
  - New parameters are introduced to work with multi-operator mode along with the above parameters. These parameters are required only when one or more operators are enabled.
  - params: `{"operatorId": y}`(for normal operator)
    - "operatorId"— operator index (Only 0, 1, 2 are valid ID numbers).
    - If user wants to use "super-operator" mode, its password is required to send as a parameter.
  - params: `{"password": "xyz"}` (for super operator)
    - password — Super user password saved. When this is passed, actions will be performed as if no other operator is enabled.
- Response:
  - response: null
  - success: (0=success, anything else is an error)
- Examples:
  - `{"params": {}, "method":"allTrans", "id":"1234", "jsonrpc":"2.0"}`
  - `{"params": {"transTime": 40}, "method":"allTrans", "id":"1234", "jsonrpc":"2.0"}`

  For normal operator:
  - `{"params": {"operatorId": 1}, "method":"allTrans", "id":"1234", "jsonrpc":"2.0"}`

  For super operator:
  - `{"params": {"password": "123"}, "method":"allTrans", "id":"1234", "jsonrpc":"2.0"}`

cut

- Definition
  - It executes the "Cut" command.
  - If multi-operator mode is enabled, cut will affect only those destinations which are selected for the requested operator.
- Request:
  - params: `{}` – None required, unless the system is using the multi-operator environment.
- Multi-Operator Mode:
  - New parameters are introduced to work with multi-operator mode along with the above parameters. These parameters are used only when one or more operators are enabled.
  - params: `{"operatorId": y}` (for normal operator)
    - "operatorId" — operator index (For current release only 0, 1, 2 are valid indices).
    - If user wants to use "super-operator" mode, its password is required which is passed as a parameter.
  - params: `{"password": "xyz"}` (for super operator)
    - `password` — Super user password saved. When this is passed, actions will be performed as if no other operator is enabled.
- Response:
  - response: null
  - success: (0=success, anything else is an error)
- Example
  - `{"params": {}, "method":"cut", "id":"1234", "jsonrpc":"2.0"}`

  For normal operator:
  - `{"params": {"operatorId": 1}, "method":"cut", "id":"1234", "jsonrpc":"2.0"}`

  For super operator:
  - `{"params": {"password": "123"}, "method":"cut", "id":"1234", "jsonrpc":"2.0"}`

listCues

- Definition:
  - List all the cues.
- Request:
  - params: `{"id": x}`
    - id — Index of the Cue.
- Response:
  - Response: Array of cue objects.
  - success: (0=success, anything else is an error)
- Example:
  - `{"params": {}, "method":"listCues", "id":"1234", "jsonrpc":"2.0"}`

activateCue

- Definition:
  - This API gives option to play/pause/stop cue.
- Request:
  - params: `{"id": 1, "type": x}, "method":"activateCue", "id":"1234", "jsonrpc":"2.0"}`
    - id — Index of the cue.
    - type — (Default is play). "x" can be: 0 — Play. 0 is the default value for the type parameter. 1 — Pause, 2 — Stop
- Response:
  - Response: null
  - success: (0=success, anything else is an error)
- Example:
  - `{"params": {"id": 1}, "method":"activateCue", "id":"1234", "jsonrpc":"2.0"}`
    //Play – no params or type 0
  - `{"params": {"type": 1}, "method":"activateCue", "id":"1234", "jsonrpc":"2.0"}`
    //Pause – type 1
  - `{"params": {"type": 2}, "method":"activateCue", "id":"1234", "jsonrpc":"2.0"}`
    //Stop – type 2

listMvrPreset

- Definition:
  - This API lists any saved MVR Presets and their included MVR Layouts per MVR ID number.
- Request:
  - params: `{"id": x}`

    "x" can be:
    - –1: Do not care (All presets). (Has priority over a particular ID, if passed as a parameter.)
    - 0–9: want to see the information associated with the MVR Preset ID number only.

- Response:
  - response: Array of: `[{"id":0-9, "Name":"mvrPresetName", "lockMode":0-1, "layoutMap": "mvrId,layoutId"}]`
    - Response contains the array of presets. Above response contains MVR Preset ID number, MVR Preset name, lock mode, layout amp contained in the preset.
  - success: (0=success, anything else is an error)
- Example:
  - `{"params": {"id": -1}, "method": "listMvrPreset", "id": "1234", "jsonrpc": "2.0"}`

activateMvrPreset

- Definition:
  - This API recalls any saved MVR Preset.
- Request:
  - params: `{"id": x}`

    "x" can be:
    - 0–9: ID number of the MVR Preset to be recalled.

- Response:
  - success: (0=success, anything else is an error)
- Example:
  - `{"params": {"id": 1}, "method": "activateMvrPreset", "id": "1234", "jsonrpc": "2.0"}`
    //Activates MVR Preset number 2

mvrLayoutChange

- Definition:
  - This API changes layout in the given frame multiviewer.
- Request:
  - params: `{"frameUnitId": x, "mvrId": x, "mvrLayoutId": x}`
    - frameUnitId: Frame unit id for which MVR layout needs to be changed.
    - mvrId: MVR ID index. Possible values are 0 and 1.
    - mvrLayoutId: MVR layout index. Possible value 0 to 9.
    - All three are mandatory parameters for this API.

- Response:
  - response: null
  - success: (0=success, anything else is an error)
- Example:
  - `{"params": {"frameUnitId": 0, "mvrId": 1, "mvrLayoutId": 3}, "method": "mvrLayoutChange", "id":"1234", "jsonrpc":"2.0"}`
    //On unit ID 0, change the second MVR's layout to number 4