



Vertiv™ Avocent® Matrix Manager Software API

Technical Bulletin

Technical Support Site

If you encounter any installation or operational issues with your product, check the pertinent section of this manual to see if the issue can be resolved by following outlined procedures. For additional assistance, visit <https://www.VertivCo.com/en-us/support/>.

TABLE OF CONTENTS

1 Overview	1
1.1 Prerequisites	1
1.1.1 Matrix manager software	1
1.1.2 REST client	1
1.1.3 Telnet authentication	2
2 RESTful API Requests	3
2.1 Login User Request	3
2.2 Logout User Request	3
2.3 Establish Connection (Session) Request	4
2.4 Disconnect Connection (Session) Request	5
2.5 Get Users List Request	6
2.6 Get Device List Request	8
2.7 Authorized Device List For User Request	12
2.8 Get Session List Request	15
2.9 Get External Target Device List Request	18
2.10 Show Message on the Receiver OSD Request	20
2.11 Get Connection Path List Request	22
3 Telnet API	29
3.1 Plain Text Format	29
3.2 JSON Format	31

1 OVERVIEW

The Matrix Manager software utilizes an application programming interface (API) service that allows you to customize or develop third party applications to interact with the Matrix Manager software. The Matrix Manager software external APIs allow administrators with an external API access role to remotely access Matrix Manager software data such as users, devices and sessions. The forced log in/log out of users into receivers as well as the establishing and disconnecting of receiver and transmitter sessions can all be done through the API interface. The External API utilizes open standards such as RESTful Web Service and Telnet. The following sections detail the URL format, parameter information and available data types to a Matrix Manager software user via these APIs.

1.1 Prerequisites

Complete the following before invoking the API.

1.1.1 Matrix manager software

The Matrix Manager software should be installed in a recommended server environment with the firewall configured to allow any incoming requests from clients. The default port for REST API is 8443, but it can be changed during the software installation. The software should also be configured to allow access for the required internal or external users, groups and roles. The default port for the Telnet API is 9000 and can be configured from the Manager software Web User Interface (UI). The devices should also be discovered and enrolled.

By default, external API access is disabled, so access must first be enabled to continue.

To enable external API access:

1. From the Dashboard side navigation bar, select *Settings – System tab – External API*. The External API pane appears.
2. For the external web access API, select *Enable* from the drop-down menu.

-or-

For Telnet access API:

- a. Select *Enable* from the drop-down menu.
 - b. Enter the Telnet access port number.
3. Click *Save*.

To access Matrix Manager software using an external API:

1. Create a user in Matrix Manager System with Manager API Access.
2. Map the user to a user role which has API access.
3. In the Web API request or when prompted in the Telenet session, enter the username and password of the user with API access.

1.1.2 REST client

The REST client must be able to accept self-signed server certificates and configured in Allow All mode for hostname verification for https calls.

For the REST requests, the headers should be entered as follows:

Content-type = application/json

Accept = application/json

If one or more optional parameters are specified in the request body, the response is based on the combination of all values.

To access the URL for the external API, enter the following with the corresponding resource name:
<https://<IP>:<Port>/matrixmanager/ext-api>.

1.1.3 Telnet authentication

In the Telnet API there are two ways to pass parameter:

- JavaScript Object Notation (JSON)
- Plain text

NOTE: If you need to filter the data using specific criteria, JSON is the preferred method.

In order to access the Manager software through the Telnet API, you will need to provide an administrator username and password with API access.

2 RESTFUL API REQUESTS

The following are requests available through the RESTful external API. The requests authenticate and authorize the user to pull data from the Manager software into third party applications.

2.1 Login User Request

The Matrix Manager software authenticates the admin user and verifies if Login User has device access to receiver. If the user has access to the receiver, the software sends a request to receiver for force login.

Table 4.1 Login User Request

PARAMETER	REQUIRED	TYPE	DESCRIPTION
adminUserName	Y	String[20]	Admin user name which has MANAGER_API_ACCESS permission
adminPassword	Y	String[40]	Admin user password for manager software side authentication
loginUserName	Y	String[20]	Login user name who will be logged in to receiver via force login. User must have DEVICE_ACCESS permission on provided receiver
rxEID	Y	String[13]	Receiver EID

Resource URL: /login

Methods: POST

Example:

Request: POST MANAGER_BASE_URL/login

Request Body:

```
{
  "adminUserName": "admin",
  "adminPassword": "123",
  "loginUserName": "asingh6",
  "rxEID": "rx1"
}
```

Response:

Status: 200 OK

2.2 Logout User Request

NOTE: Login user must be an admin user and have user session in Matrix Manager system with the specified receiver.

The Matrix Manager software authenticates the admin user and verifies any user session exists in system with provided login user and receiver EID. If verification is successful, the manager software sends a request to the receiver to log out the user from the device and the manager software deletes the user sessions from the system.

Table 4.2 Logout User Request

PARAMETER	REQUIRED	TYPE	DESCRIPTION
adminUserName	Y	String[20]	Admin user name which has MANAGER_API_ACCESS permission
adminPassword	Y	String[40]	Admin user password for manager software side authentication
loginUserName	Y	String[20]	Login user name who will be logged in to receiver via force login. User must have DEVICE_ACCESS permission on provided receiver
rxEID	Y	String[13]	Receiver EID

Resource URL: /logout

Methods: POST

Example:

```
Request: POST MANAGER_BASE_URL/logout
Request Body:
{
  "adminUserName":"admin",
  "adminPassword":"123",
  "loginUserName":"asingh6",
  "rxEID":"rx1"
}
Response:
Status: 200 OK
```

2.3 Establish Connection (Session) Request

NOTE: Login user must be an admin user and have a user session in the Matrix Manager system with specified receiver.

The Matrix Manager software authenticates the admin user and verifies any user session exists in system with provided login user and receiver EID. The Matrix Manager software verifies if login user has device access permission on provided transmitter EID. If verification is successful, the software sends a request to establish a connection between the receiver and transmitter in the provided mode. The software then updates the connection details on the system.

Table 4.3 Establish Connection Request

PARAMETER	REQUIRED	TYPE	VALUES	DESCRIPTION
adminUserName	Y	String[20]	*	Admin user name which has MANAGER_API_ACCESS permission
adminPassword	Y	String[40]	*	Admin user password for manager software side authentication
loginUserName	Y	String[20]	*	Login user name who will be logged in to receiver via force login. User must have DEVICE_ACCESS permission on provided receiver
rxEID	Y	String[13]	*	Receiver EID
txEID	Y	String[13]	*	Transmitter EID
modeAllowed	Y	Enum	share_mode, private_mode	Connection mode

NOTE: *Values based on the parameter.

Resource URL: / connect

Methods: POST

Example:

```
Request: POST MANAGER_BASE_URL/ connect
Request Body:
{
  "adminUserName":"admin",
  "adminPassword":"123",
  "loginUserName":"asingh6",
  "rxEID":"rx1",
  "txEID":"tx1",
  "modeAllowed":"share_mode"
}
Response:
Status: 200 OK
```

2.4 Disconnect Connection (Session) Request

NOTE: Login user must be and admin user and have a user session in the Matrix Manager system with the specified receiver.

The Matrix Manager software authenticates the admin user and verifies any user session exists in system with the provided login user and receiver EID. The software also verifies if a connection exists with user session. If verification is successful, the software sends the disconnect request to device.

Table 4.4 Disconnect Session Request

PARAMETER	REQUIRED	TYPE	DESCRIPTION
adminUserName	Y	String[20]	Admin user name which has MANAGER_API_ACCESS permission
adminPassword	Y	String[40]	Admin user password for manager software side authentication
loginUserName	Y	String[20]	Login user name who will be logged in to receiver via force login. User must have DEVICE_ACCESS permission on provided receiver
rxEID	Y	String[13]	Receiver EID
txEID	Y	String[13]	Transmitter EID

Resource URL: / disconnect

Methods: POST

Example:

```
Request: POST MANAGER_BASE_URL/ disconnect
Request Body:
{
  "adminUserName":"admin",
  "adminPassword":"123",
  "loginUserName":"asingh6",
  "rxEID":"rx1",
  "txEID" : "tx1"
}
Response:
Status: 200 OK
```

2.5 Get Users List Request

The API can be used to retrieve a list of all users in the Matrix Manager system with their username, first name, last name and user description information. If the optional username field is specified in the request, then the details of the user account corresponding to the given username is returned.

Table 4.5 Get Users List Request

PARAMETER	REQUIRED	TYPE	DESCRIPTION
adminUserName	Y	String	Admin user name which has MANAGER_API_ACCESS permission
adminPassword	Y	String	Admin user password for manager software side authentication
userName	N	String	Optional username of the user account to be retrieved from Matrix manager. (If there is no username specified, and then all users from Matrix manager will be returned in response).

Table 4.6 Get Users List Response

PARAMETER	REQUIRED	TYPE	DESCRIPTION
userName	Y	String	User account name
firstName	Y	String	User account first name
lastName	Y	String	User account last name
description	Y	String	User description

Resource URL: MANAGER_BASE_URL/getUsers

Methods: POST

Example: Get specific user account information.

Request Body:

```
{
  "adminUserName": "roger",
  "adminPassword": "123456",
  "userName": "roger"
}
```

Response: Status: 200 OK

```
[
  {
    "userName" : "roger",
    "firstName" : "roger",
    "lastName" : "liu",
    "description" : "System Administrator."
  }
]
```

Example: Get all user accounts list from Matrix Manager software.

```
Request Body:
{
  "adminUserName":"roger",
  "adminPassword":"123456"
}
Response: Status: 200 OK
[
  {
    "userName" : "roger",
    "firstName" : "roger",
    "lastName" : "liu",
    "description" : "System Administrator.",
  },
  {
    "userName" : "Admin",
    "firstName" : "Matrix",
    "lastName" : "Admin",
    "description" : "System Default Admin User.",
  }
]
```

2.6 Get Device List Request

The API can be used to retrieve the list of all devices in the Matrix Manager system. If one or more of the optional parameters (device name, EID or type) are specified in the request body, then information about devices matching the specified criteria are returned.

Table 4.7 Get Device List Request

PARAMETER	REQUIRED	TYPE	VALUES	DESCRIPTION
adminUserName	Y	String	*	Admin user name which has MANAGER_API_ACCESS
adminPassword	Y	String	*	Admin user password for manager software side authentication
deviceName	N	String	*	Optional device name parameter to obtain details of a particular device
eid	N	String	*	Optional EID parameter to obtain details of a particular device
deviceType	N	String	rx,tx,swx,	Optional device type parameter to obtain a list of particular type of devices.
system	N	String	Matrix, HMX, AMX	NOTE: These values in this table represent the Avocent® HMX Manager software or the Avocent® AMX™ switching system. Optional parameter to query devices from a particular system

NOTE: *Values based on the parameter.

Table 4.8 Get Device List Response

PARAMETER	REQUIRED	TYPE	VALUES	DESCRIPTION
deviceName	Y	String	*	Device name
type	Y	String	rx,tx,swx,	Device type
eid	Y	String	*	Device EID
status	Y	String	*	Current status of the device
ipv4Address	N	String	*	IPV4 address for switch
ipv6Address	N	String	*	IPV6 address for switch
system	Y	String	Matrix, HMX, AMX	System to which this device belongs to
deviceUid	N	String	*	Device UID (only for external KVM systems)

Resource URL: `MANAGER_BASE_URL/getDevices`

Methods: POST

Example: Get specific device (receiver/transmitter) information based on EID.

Request Body:

```
{
  "adminUserName":"roger",
  "adminPassword":"123456",
  "eid":"510162-E00004"
}
```

Response: Status: 200 OK

```
[
  {
    "deviceName" : "MXR5110-RX-DVI-I-E00004",
    "type" : "rx",
    "eid" : "510162-E00004",
    "status" : "ready",
    "system" : "Matrix"
  }
]
```

Example: Get specific device (switch) information based on device name.

Request Body:

```
{  
  "adminUserName": "roger",  
  "adminPassword": "123456",  
  "deviceName": "MXS5132-SX-32-3032C6"  
}
```

Response: Status: 200 OK

```
[  
  {  
    "deviceName" : "MXS5132-SX-32-3032C6",  
    "type" : "swx",  
    "eid" : "520EMR-3032C6",  
    "ipv4Address" : "10.207.56.120",  
    "ipv6Address" : "fe80::217:ebff:fe30:32c6",  
    "status" : "ready",  
    "system" : "Matrix"  
  }  
]
```

Example: Get specific device (receiver) information based on device type.

Request Body:

```
{  
  "adminUserName": "roger",  
  "adminPassword": "123456",  
  "deviceType": "rx"  
}
```

Response: Status: 200 OK

```
[  
  {  
    "deviceName" : "MXR5110-RX-DVI-I-E05103",  
    "type" : "rx",  
    "eid" : "510162-E05103",  
    "status" : "ready",  
    "system" : "Matrix"  
  },  
  {  
    "deviceName" : "012F121203134044",  
    "type" : "rx",  
    "eid" : "610162-1BIXT4",  
    "system" : "AMX",  
    "deviceUid" : "012F121203134044"  
  }  
]
```

Example: Get full device list from Matrix Manager software.

```
Request Body:
{
  "adminUserName":"roger",
  "adminPassword":"123456"
}
Response: Status: 200 OK
[
  {
    "deviceName" : "MXT5110-TX-DVI-D-E04103",
    "type" : "tx",
    "eid" : "500200-E04103",
    "status" : "ready",
    "system" : "Matrix"
  },
  {
    "deviceName" : "MXS5132-SX-32-3032C6",
    "type" : "swx",
    "eid" : "520EMR-3032C6",
    "ipv4Address" : "10.207.56.120",
    "ipv6Address" : "fe80::217:ebff:fe30:32c6",
    "status" : "ready",
    "system" : "Matrix"
  },
  {
    "deviceName" : "MXR5110-RX-DVI-I-E05139",
    "type" : "rx",
    "eid" : "510162-E05139",
    "status" : "ready",
    "system" : "Matrix"
  }
]
```

2.7 Authorized Device List For User Request

The API provides a mechanism to retrieve the list of authorized devices for the given user name. An optional device type parameter can be specified in the request to query for authorized devices of a particular type.

Table 4.9 Authorized Device List For User Request

PARAMETER	REQUIRED	TYPE	VALUES	DESCRIPTION
adminUserName	Y	String	*	Admin user name which has MANAGER_API_ACCESS
adminPassword	Y	String	*	Admin user password for manager side authentication
userName	Y	String	*	userName for which the list of authorized devices is retrieved.
deviceType	N	String	rx,tx,swx	Optional deviceType parameter to retrieve authorized list of devices of a particular type alone

NOTE: *Values based on the parameter.

Table 4.10 Authorized Device List For User Response

PARAMETER	REQUIRED	TYPE	VALUES	DESCRIPTION
deviceName	Y	String	*	Device name
type	Y	String	rx,tx,swx,	Device type
eid	Y	String	*	Device EID
status	N	String	*	Current status of the device
ipv4Address	N	String	*	IPV4 address for switch
ipv6Address	N	String	*	IPV6 address for switch
system	Y	String	Matrix, HMX, AMX	NOTE: These values in this table represent the Avocent® HMX Manager software or the Avocent® AMX™ switching system. Identifier to show this device belongs to which system.
deviceUid	N	String	*	Device UID (only for External KVM devices)

NOTE: *Values based on the parameter.

Resource URL: [MANAGER_BASE_URL/getDevicesByUser](#)

Methods: POST

Example: Get authorized device list for user (switches/receivers/transmitters).

Request Body:

```
{
  "adminUserName": "roger",
  "adminPassword": "123456",
  "userName": "charlie"
}
```

Response: Status: 200 OK

```
[
  {
    "deviceName": "MXR5110-RX-DVI-I-E00004",
    "type": "rx",
    "eid": "510162-E00004",
    "status": "ready",
    "system": "Matrix"
  },
  {
    "deviceName": "MXS5132-SX-32-3032C6",
    "type": "swx",
    "eid": "520EMR-3032C6",
    "ipv4Address": "10.207.56.120",
    "ipv6Address": "fe80::217:ebff:fe30:32c6",
    "status": "ready",
    "system": "Matrix"
  },
  {
    "deviceName": "MXT5110-TX-DVI-D-E04022",
    "type": "tx",
    "eid": "500200-E04022",
    "status": "ready",
    "system": "Matrix"
  },
  {
    "deviceName": "MXR5110-RX-DVI-I-E00003",
    "type": "rx",
    "eid": "510162-E00003",
    "status": "ready",
    "system": "Matrix"
  },
  {
    "deviceName": "010F10025426",
    "type": "tx",
    "system": "AMX",
    "eid": "600200-PHCRTY"
  },
  {
    "deviceName": "DBotRT_5100PT12",
```

```

    "type" : "tx",
    "system" : "AMX",
    "eid" : "600200-P7INRU"
  }
]

```

Example: Get authorized device list for user based on device type.

Request Body:

```

{
  "adminUserName": "roger",
  "adminPassword": "123456",
  "userName": "charlie",
  "deviceType": "tx"
}

```

Response: Status: 200 OK.

```

[
  {
    "deviceName" : "MXT5110-TX-DVI-D-E04022",
    "type" : "tx",
    "eid" : "500200-E04022",
    "status" : "ready",
    "system" : "Matrix"
  },
  {
    "deviceName" : "010F10025426",
    "type" : "tx",
    "system" : "AMX",
    "eid" : "600200-PHCRTY"
  },
  {
    "deviceName" : "DBotRT_5100PT12",
    "type" : "tx",
    "system" : "AMX",
    "eid" : "600200-P7INRU"
  }
]

```

2.8 Get Session List Request

The API can be used to retrieve the list of all currently active receiver logins and connections between receivers and transmitters. Based on the optional (rxEid/rxName/txEid/txName) parameter provided in the request body, the list can be filtered to obtain specific session information.

Table 4.11 Get Session List Request

PARAMETER	REQUIRED	TYPE	DESCRIPTION
adminUserName	Y	String	Admin user name which has MANAGER_API_ACCESS
adminPassword	Y	String	Admin user password for manager software side authentication
rxName	N	String	Receiver name
rxEid	N	String	Receiver EID
txName	N	String	Transmitter name
txEid	N	String	Transmitter EID

Table 4.12 Get Session List Response

PARAMETER	REQUIRED	TYPE	VALUE	DESCRIPTION
rxName	Y	String	*	Receiver name
rxEid	Y	String	*	Receiver EID
txName	N	String	*	Transmitter name
txEid	N	String	*	Transmitter EID
bridgeRxName	N	String	*	External bridge receiver name only showed if this connection has foreign target connected
bridgeRxEid	N	String	*	External receiver EID which was used as the bridge connector.
bridgeTxName	N	String	*	Matrix bridge transmitter name used to create foreign connection
bridgeTxEid	N	String	*	Matrix bridge transmitter EID (For external connections only)
establishTime	Y	String	*	Time at which the connection was created
userName	Y	String	*	Username of the user logged in to the Matrix receiver
mode	N	String	share_mode, private_mode	Connection mode
status	N	String	*	Connection status

NOTE: *Values based on the parameter.

Resource URL: MANAGER_BASE_URL/getSessions

Methods: POST

Example: Get connections based on receiver/transmitter EID.

Request Body:

```
{
  "adminUserName":"roger",
  "adminPassword":"123456",
  "rxEid":"510162-E00003"
}
```

Response: Status: 200 OK

```
[
  {
    "rxName" : "MXR5110-RX-DVI-I-E00003",
    "rxEid" : "510162-E00003",
    "txName" : "010F10025426",
    "txEid" : "600200-WJX8P3",
    "bridgeRxName" : "012F121161324046",
    "bridgeRxEid" : "610162-D5S1XV",
    "bridgeTxName" : "MXT5120-TX-VGA-E01011",
    "bridgeTxEid" : "500201-E01011",
    "establishTime" : "2014-06-06 14:16:03.144",
    "userName" : "Admin",
    "mode" : "private_mode",
    "status" : "connected"
  }
]
```

Example: Get connections based on receiver/transmitter name.

Request Body:

```
{
  "adminUserName":"roger",
  "adminPassword":"123456",
  "txName":"MXT5110-TX-DVI-D-E04022"
}
```

Response: Status: 200 OK.

```
[
  {
    "rxName" : "MXR5110-RX-DVI-I-E05139",
    "rxEid" : "510162-E05139",
    "txName" : "MXT5110-TX-DVI-D-E04022",
    "txEid" : "500200-E04022",
    "establishTime" : "2014-06-06 14:16:43.347",
    "userName" : "Admin",
    "mode" : "share_mode",
    "status" : "connected"
  }
]
```

Example: Get connections without any parameter.

Request Body:

```
{
  "adminUserName": "roger",
  "adminPassword": "123456"
}
```

Response: Status: 200 OK.

```
[
  {
    "rxName" : "MXR5110-RX-DVI-I-E05139",
    "rxEid" : "510162-E05139",
    "establishTime" : "2014-06-12 10:22:59.481",
    "userName" : "Admin"
  },
  {
    "rxName" : "MXR5110-RX-DVI-I-E00003",
    "rxEid" : "510162-E00003",
    "txName" : "010F10025426",
    "txEid" : "600200-WJX8P3",
    "bridgeRxName" : "012F121161324046",
    "bridgeRxEid" : "610162-D5S1XV",
    "bridgeTxName" : "MXT5120-TX-VGA-E01011",
    "bridgeTxEid" : "500201-E01011",
    "establishTime" : "2014-06-11 10:34:29.612",
    "userName" : "Admin",
    "mode" : "private_mode",
    "status" : "connected"
  },
  {
    "rxName" : "MXR5110-RX-DVI-I-E00004",
    "rxEid" : "510162-E00004",
    "txName" : "MXT5110-TX-DVI-D-E04022",
    "txEid" : "500200-E04022",
    "establishTime" : "2014-06-11 10:43:34.706",
    "userName" : "Admin",
    "mode" : "share_mode",
    "status" : "connected"
  }
]
```

2.9 Get External Target Device List Request

The API can be used to retrieve all external target devices managed by the software. Based on the optional system parameter, target devices under a particular system alone can be retrieved.

Table 4.13 Get External Target Device List Request

PARAMETER	REQUIRED	TYPE	VALUES	DESCRIPTION
adminUserName	Y	String	*	Admin user name which has MANAGER_API_ACCESS
adminPassword	Y	String	*	Admin user password for manager software side authentication
system	N	String	AMX, HMX	NOTE: These values in this table represent the Avocent® HMX Manager software or the Avocent® AMX™ switching system. This parameter to define the external target device comes from which external system.

NOTE: *Values based on the parameter.

Table 4.14 Get External Target Device List Response

PARAMETER	REQUIRED	TYPE	VALUES	DESCRIPTION
deviceName	Y	String	*	External target name
eid	Y	String	*	External target EID
deviceUid	Y	String	*	External target UID from external system
system	Y	String	AMX, HMX	NOTE: These values in this table represent the Avocent® HMX Manager software or the Avocent® AMX™ switching system. System corresponding to the external device.

NOTE: *Values based on parameter.

Resource URL: MANAGER_BASE_URL/getExternalTargetDevices

Methods: POST

Example: Get external target device based on external system.

```

Request Body:
{
  "adminUserName":"roger",
  "adminPassword":"123456",
  "system":"AMX"
}
Response: Status: 200 OK
[
  {
    "deviceName" : "DBotRT_5100PT12",
    "eid" : "600200-KUEOVE",
    "deviceUid" : "012A0E0207061010",
    "system" : "AMX"
  }
]

```

Example: Get full list of external target devices from Matrix system.

```

Request Body:
{
  "adminUserName":"roger",
  "adminPassword":"123456"
}
Response: Status: 200 OK.
[
  {
    "deviceName" : "010F10025426",
    "eid" : "600200-5CGAL2",
    "deviceUid" : "010F100254261010",
    "system" : "AMX"
  },
  {
    "deviceName" : "DBotRT_5100PT12",
    "eid" : "600200-KUEOVE",
    "deviceUid" : "012A0E0207061010"
    "system" : "AMX"
  }
]

```

2.10 Show Message on the Receiver OSD Request

The API can be used to send a message to be displayed on a receiver OSD. Same or different messages can be sent to multiple receivers using a single request. Each message is sent as a combination of type, text and receiver EID within the request.

Table 4.15 OSD Receiver Message Request

PARAMETER	REQUIRED	TYPE	VALUES	DESCRIPTION
adminUserName	Y	String	*	Admin user name which has MANAGER_API_ACCESS
adminPassword	Y	String	*	Admin user password for manager software side authentication
messages	Y	List<message>	*	An array of messages object

NOTE: *Values based on the parameter.

Table 4.16 Receiver Messages

PARAMETER	REQUIRED	TYPE	VALUES	DESCRIPTION
type	Y	String	info, warning, error	Type of the message.
text	Y	String[120]	*	The actual text to be displayed in receiver OSD.
rxEid	Y	String	*	EID of the receiver to which this message is sent

NOTE: *Values based on the parameter.

Resource URL: MANAGER_BASE_URL/showMessage

Methods: POST

Example: Get external target device based on external system.

Request Body:

```
{
  "adminUserName":"roger",
  "adminPassword":"123456",
  "messages":
  [
    {
      "type":"info",
      "text":"How are you doing today!",
      "rxEid":"510162-E05139"
    },
    {
      "type":"warning",
      "text":"You are not authorized to access this target!",
      "rxEid":"510162-E05140"
    },
    {
      "type":"error",
      "text":"The device you trying to access is no longer exist!",
      "rxEid":"510162-E05141"
    }
  ]
}
```

Response: Status: 200 OK

If there is a problem in trying to display one or more message sent in the request, the response would contain the receiver EID where the failure occurred and the corresponding failure message.

Example: Failure response.

```
{
  [
    {
      "rxEid": "510162-E05139",
      "errors" : [
        {
          "ermo": "80",
          "details": "failed to communicate to receiver"
        }
      ]
    }
  ]
}
```

2.11 Get Connection Path List Request

The API can be used to retrieve the list of all currently active connections path between receivers and transmitters. Based on the optional (rxEid/rxName/txEid/txName) parameter provided in the request body, the list can be filtered to obtain specific connection information. The response represents the connection path in the order from transmitter to the receiver including bridge information for connections to external targets.

Table 4.17 Get ConnectionPath List Request

PARAMETER	REQUIRED	TYPE	DESCRIPTION
adminUserName	Y	String	Admin user name which has MANAGER_API_ACCESS
adminPassword	Y	String	Admin user password for manager software side authentication
rxName	N	String	Receiver name
rxEid	N	String	Receiver EID
txName	N	String	Transmitter name
txEid	N	String	Transmitter EID

Table 4.18 Get ConnectionPath List Response

PARAMETER	REQUIRED	TYPE	DESCRIPTION
rxName	Y	String	Receiver name
rxEid	Y	String	Receiver EID
txName	Y	String	Transmitter name
txEid	Y	String	Transmitter EID
bridgeRxName	N	String	External bridge receiver name (Only showed if this connection has foreign target connected)
bridgeRxEid	N	String	External receiver EID used as the bridge connector
bridgeTxName	N	String	Matrix bridge transmitter name (used to create foreign connection)
bridgeTxEid	N	String	Matrix bridge transmitter EID (For external connections only)
connectionPath	Y	List<connectionPath>	An array of connectionPath object

Table 4.19 Object ConnectionPath

PARAMETER	REQUIRED	TYPE	DESCRIPTION
switchName	Y	String	Switch name used in the connection
switchEid	Y	String	Switch EID used in the connection
portIn	Y	String	Video flow port in
portOut	Y	String	Video flow port out

Resource URL: `MANAGER_BASE_URL/getConnectionPath`

Methods: POST

Example: Get connections based on receiver/transmitter EID.

Request Body:

```
{
  "adminUserName": "Admin",
  "adminPassword": "",
  "rxEid": "510162-E00004"
}
```

Response: Status: 200 OK

```
[
  {
    "txName": "010F10025426",
    "txEid": "600200-WJX8P3",
    "bridgeRxName": "012F121203134044",
    "bridgeRxEid": "610162-1BIXT4",
    "bridgeTxName": "MXT5120-TX-VGA-000095",
    "bridgeTxEid": "500201-000095",
    "connectionPath": [
      {
        "switchName": "MXS5132-SX-32-3032C6",
        "switchEid": "520EMR-3032C6",
        "portIn": "5",
        "portOut": "11"
      }
    ],
    "rxName": "MXR5110-RX-DVI-I-E00004",
    "rxEid": "510162-E00004"
  }
]
```

Example: Get connections based on receiver/transmitter name.

```

Request Body:
{
  "adminUserName":"Admin",
  "adminPassword": "",
  "txName" : "MXT5110-TX-DVI-D-E04103"
}
Response: Status: 200 OK.
[
  {
    "txName" : "MXT5110-TX-DVI-D-E04103",
    "txEid" : "500200-E04103",
    "connectionPath" : [
      {
        "switchName" : "MXS5132-SX-32-3032C6",
        "switchEid" : "520EMR-3032C6",
        "portIn" : "21",
        "portOut" : "13"
      }
    ],
    "rxName" : "MXR5110-RX-DVI-I-E00003",
    "rxEid" : "510162-E00003"
  },
  {
    "txName" : "MXT5110-TX-DVI-D-E04103",
    "txEid" : "500200-E04103",
    "connectionPath" : [
      {
        "switchName" : "MXS5132-SX-32-3032C6",
        "switchEid" : "520EMR-3032C6",
        "portIn" : "21",
        "portOut" : "7"
      }
    ],
    "rxName" : "MXR5110-RX-DVI-I-E05139",
    "rxEid" : "510162-E05139"
  }
]

```

Example: Get connections without any parameter.

```

Request Body:
{
  "adminUserName":"roger",
  "adminPassword":"123456"
}

```

Response: Status: 200 OK.

```
[
  {
    "txName" : "MXT5110-TX-DVI-D-E04103",
    "txEid" : "500200-E04103",
    "connectionPath" : [
      {
        "switchName" : "MXS5132-SX-32-3032C6",
        "switchEid" : "520EMR-3032C6",
        "portIn" : "21",
        "portOut" : "13"
      }
    ]
  },
  {
    "rxName" : "MXR5110-RX-DVI-I-E00003",
    "rxEid" : "510162-E00003"
  },
  {
    "txName" : "010F10025426",
    "txEid" : "600200-WJX8P3",
    "bridgeRxName" : "012F121203134044",
    "bridgeRxEid" : "610162-1BIXT4",
    "bridgeTxName" : "MXT5120-TX-VGA-000095",
    "bridgeTxEid" : "500201-000095",
    "connectionPath" : [
      {
        "switchName" : "MXS5132-SX-32-3032C6",
        "switchEid" : "520EMR-3032C6",
        "portIn" : "5",
        "portOut" : "11"
      }
    ]
  },
  {
    "rxName" : "MXR5110-RX-DVI-I-E00004",
    "rxEid" : "510162-E00004"
  },
  {
    "txName" : "MXT5110-TX-DVI-D-E04103",
    "txEid" : "500200-E04103",
    "connectionPath" : [
      {
        "switchName" : "MXS5132-SX-32-3032C6",
        "switchEid" : "520EMR-3032C6",
        "portIn" : "21",
        "portOut" : "7"
      }
    ]
  },
  {
    "rxName" : "MXR5110-RX-DVI-I-E05139",
    "rxEid" : "510162-E05139"
  }
]
```

]

This page intentionally left blank.

3 TELNET API

The following section provides an overview of the Manager software Telnet API calls.

3.1 Plain Text Format

The following table lists the plain text based format commands. The order of parameters must be maintained.

Table 5.1 Plain Text Commands

METHOD	PARAMETERS	FORMAT	EXAMPLE
help	command (optional)	<p>Command to find all commands available in the API and their help messages.</p> <p>Command format: help [command]</p> <p>Command: command to show help for any particular command in system</p>	<p>help</p> <p>help osdlogin</p>
osdlogin	<p>rxEID (mandatory)</p> <p>loginUserName (mandatory)</p>	<p>Command to invoke force login to receiver On Screen Display (OSD)</p> <p>Command format: osdlogin rxEID loginUserName</p> <p>rxEID: receiver EID</p> <p>loginUserName: login userName</p>	osdlogin 510162-E05046 Admin
osdlogout	rxEID (mandatory)	<p>Command to invoke force logout from receiver OSD</p> <p>Command format: osdlogout rxEID</p> <p>rxEID: receiver EID</p>	osdlogout 510162-E05038
connect	<p>rxEID (mandatory)</p> <p>txEID (mandatory)</p> <p>mode (mandatory)</p>	<p>Command to invoke force connect from given receiver</p> <p>Command format:</p> <p>connect rxEID txEID mode</p> <p>rxEID: receiver EID</p> <p>txEID: trasmitter EID</p> <p>mode: connection mode, valid values are private/share</p>	<p>connect 510162-E05046 500200-E04036 share</p> <p>connect 510162-E00005 500200-E04036 private</p>
disconnect	rxEID (mandatory)	rxEID (mandatory)	disconnect 510162-E05038
receivers	n/a	<p>Command to get all receiver details from the Matrix Manager software</p> <p>Command format: receivers</p>	receivers
sessions	txEID (optional, filter)	<p>Command to get all user sessions and connections from Matrix Manager software</p> <p>System can filter the above list by specifying the target or transmitter EID in command</p> <p>Command format: sessions</p> <p>[txEID] txEID: [transmitter EID is optional for filtering]</p>	<p>Sessions</p> <p>Sessions 500200-E04036</p>
targets	userName (optional, filter)	<p>Command to get all transmitters from Matrix Manager software</p> <p>System can filter the authorized targets by specifying the userName in command</p> <p>Command format: targets [userName]</p> <p>userName: Optional field used for filtering authorized targets for given user</p>	<p>targets</p> <p>targets Admin</p>

METHOD	PARAMETERS	FORMAT	EXAMPLE
osdmmessage	rxEID (mandatory) msgType (mandatory) msgText (mandatory)	Command to send messages to receiver OSD Command format: osdmmessage rxEID msgType msgText rxEID: receiver EID where message will sent to be display msgType: value should be from info/warning/error msgText: message must be bounded with double quote	osdmmessage 510162-E00005 info "hello test message"
who	n/a	Command to find out what admin user logged in to access the Manager software through Telnet client	who
exit	n/a	Command to exit from Telnet session	exit

3.2 JSON Format

The following table lists the JSON based format commands.

Table 5.2 JSON Commands

METHOD	PARAMETER	DESCRIPTION	EXAMPLE
logout	n/a	Command to logout session for logged in admin user. After successful transition, system will ask to enter new admin user.	logout
exit	n/a	Command to close Telnet client session.	exit
getUsers	userName (filter, optional)	Command to get all users details in Matrix Manager software. Command can filter out user details by providing some parameter mentioned in Parameters column.	getUsers getUsers '{"userName":"asingh6"}'
getSessions	rxEID (filter, optional) rxName (filter, optional) txEID (filter, optional) txName (filter, optional)	Command to get all user sessions and connections details. Command can filter out session details by providing some parameter mentioned in Parameters column. All parameter/filter details: rxName (no) receiver name rxEid (no) receiver EID specify atleast one from rxName and rxEID txName (no) transmitter name txEid (no) transmitter EID specify at least one from txName and txEID	getSessions getSessions '{"rxEid":"RX674533","txEid":"TX1234"}' getSessions '{"rxEid":"RX674533"}'
showMessage	type (mandatory, info/warning/error) rxEID (mandatory) text (mandatory)	Command to send messages to receiver OSD. Message information will be sent in JSON format. Since system supports for multiple messages, single message also will be wrapped in JSON array format. Each message has following parameter specified as (yes/no). type (yes) message type options text (yes) message text to be sent rxEID (yes) destination receiver EID to be sent	showMessage '{"messages":[{"type":"warning", "text":"message to be sent", "rxEid":"RX674533"}]}'
getDevices	deviceName (filter, optional) EID (filter, optional) userName (filter,	Command can filter out device details by providing some parameter mentioned in Parameters column. Command to get all devices	getDevices getDevices '{"deviceName":"TestDevice","system":"MATRIX"}' getDevices '{"system":"AMX"}'

METHOD	PARAMETER	DESCRIPTION	EXAMPLE
	optional) deviceType(filter, optional) system (filter, optional)	details and filter the device list with providing some parameter (filter). Details for filters: deviceName (no) device name of device EID(no) devices userName (no) userName of user to show authorized device for user deviceType (no) device type options tx/rx/swx system (no) system type MATRIX/AMX/HMX	
getConnectionPath	rxEid txEid	Command to get connection path between given receiver and transmitter. All parameters which are required specified as (yes/no). rxName (no) receiver name rxEid (no) receiver EID specify at least one from rxName and rxEID txName (no) transmitter name txEid (no) transmitter's EID specify at least one from txName and txEID	getConnectionPath {"rxEid": "RX674533", "txEid": "TX1234"}
getDevicesByUser	userName(mandatory)	Command to get all device details by given username. All parameters which are required specified as (yes/no). userName (yes) username of user	getDevicesByUser {"userName": "asingh6"}
forceLogin	loginUserName (mandatory) rxEID(mandatory)	Command to invoke force login to receiver OSD . All parameters which are required specified as (yes/no). loginUserName (yes) login user Name which has device access rxEID (yes) receiver EID where user logs in	forceLogin {"loginUserName": "asingh6", "rxEID": "RX674533"}
forceLogout	loginUserName (mandatory) rxEID(mandatory)	Command to invoke force logout from receiver OSD. All parameters which are required specified as (yes/no). loginUserName (yes) login user name which has device access rxEID (yes)	forceLogout {"loginUserName": "asingh6", "rxEID": "RX674533"}

METHOD	PARAMETER	DESCRIPTION	EXAMPLE
		receiver EID where user logs in	
forceConnect	loginUserName (mandatory) rxEID(mandatory) txEID(mandatory) modeAllowed (mandatory)	Command to invoke force connect from given receiver to transmitter. All parameters which are required specified as (yes/no). loginUserName (yes) login user name which has device access rxEID (yes) receiver EID where login user will log in txEID (yes) transmitter EID modeAllowed (yes) share_mode/private_mode	<pre>forceConnect " {"loginUserName":"asingh6","rxEID":"RX674533","txEID":"TX1234", "modeAllowed": "share_mode"}"</pre>
forceDisconnect	loginUserName (mandatory) rxEID(mandatory) txEID(mandatory)	Command to invoke force connect from given receiver to transmitter. All parameters which are required specified as (yes/no). loginUserName (yes) login user name which has device access rxEID (yes) receiver EID where login user will log in txEID (yes) transmitter EID	<pre>forceDisconnect " {"loginUserName":"asingh6","rxEID":"RX674533","txEID":"TX1234"}"</pre>



VertivCo.com | Vertiv Headquarters, 1050 Dearborn Drive, Columbus, OH, 43085, USA

© 2017 Vertiv Co. All rights reserved. Vertiv and the Vertiv logo are trademarks or registered trademarks of Vertiv Co. All other names and logos referred to are trade names, trademarks or registered trademarks of their respective owners. While every precaution has been taken to ensure accuracy and completeness herein, Vertiv Co. assumes no responsibility, and disclaims all liability, for damages resulting from use of this information or for any errors or omissions. Specifications are subject to change without notice.

590-1740-501A