

A. IP Control Commands

1. Notice

This section is provided for informational purposes only, and should only be used by software developers with a thorough understanding of the HTTP and JSON specifications.

2. Introduction

The HDMI 4x4 Matrix Switch can be controlled using basic IP commands. These commands are based on the JSON format and are sent and received in standard TCP/IP packets. To learn more about JSON, visit <http://www.json.org>.

Only four types of JSON arrays are used to control the product:

Array of a single integer: [2]

Array of multiple integers: [1,2,3,4]

Array of a single string: ["A Name"]

Array of multiple strings: ["Name 1", "Name 2", "Name 3", "Name 4"]

Any other notation is not permitted. Also, "null" values are not permitted. Use 0 instead.

3. Basic Usage

All read and write operations are performed via HTTP GET and HTTP POST commands, respectively. While you can perform a GET at any time, a POST command will require prior authentication.

4. Basic Authentication

Authentication is performed in two simple steps:

1. Obtain a Session ID

Perform a **GET** request on **/var/session.json** to obtain a new session ID. The return value will be a JSON array of a single integer, for example **[12345]**. From then on, simply append the session to any new IP requests to use this session, e.g., **GET /var/conn.json?sid=12345**

2. To Open a Session ID that will not expires

Perform a **GET** request on **/var/xsession.json** to obtain a session ID **that will not expires**. The return value will be a JSON array of a single integer, for example **[12345]**. From then on, simply append the session to any new IP requests to use this session, e.g., **GET /var/conn.json?sid=12345**

3. Obtain Credentials for that Session ID

There are three permission levels that can be used to read/modify properties. These are:

- 0 – Guest: Can read all values. Cannot write or make changes.
- 1 – User: Same as Guest, but can change matrix video connections.
- 2 – Admin: Can perform any command without restriction.

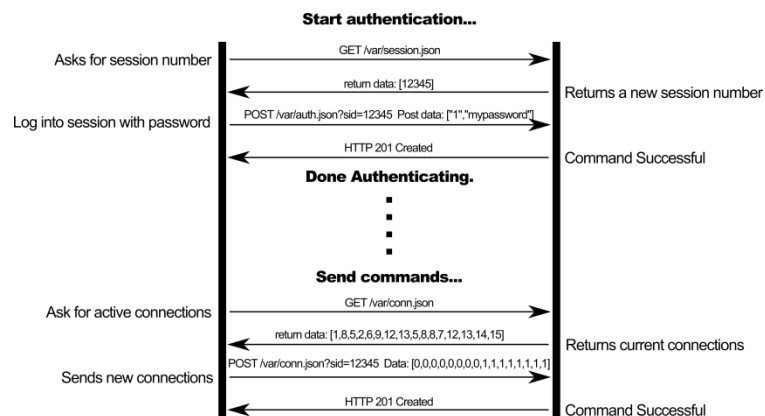
To obtain credentials, you will need to POST an array of two strings to **/var/auth.json**: the userlevel, and the password associated to that user. For example:

POST /var/auth.json?sid=12345

POST data: **[“2”, “mypassword”]**

If successful, you will receive an **HTTP 201 Created** response.

Sessions will automatically close after 5 minutes of inactivity. They may also be closed at any time by a **GET /var/logout.json**.



5. POST Commands

POST commands will always return 1 of 5 HTTP headers:

HTTP 201 Created	Command was successful
HTTP 400 Bad Request	Received argument was malformed
HTTP 403 Forbidden	User does not have sufficient privileges
HTTP 404 Not Found	URL is incorrect
HTTP 501 Not Implemented	URL does not support receiving POST data

URL	Argument
/var/conn.json	Array of connections. (Same as GET above) Example: [0,1,2,3]
/var/connio.json	Single connection JSON data= [input#,output#] Example: [2,3] (to connect input #2 to output #3)
/var/pre1.json /var/pre2.json /var/pre3.json /var/pre4.json /var/pre5.json /var/pre6.json /var/pre7.json /var/pre8.json	Array of connections for a preset (Same as GET above) Example: [0,1,2,3]
/var/boxname.json	Array of a single string. Contains the name of the box, limited to 20 characters Example: ["Classroom B Matrix"]
/var/sources.json	Array of 4 strings. Contains the names of all 4 input ports. Example: ["Cable Box", "Satellite Receiver", "Demo Reel", ...]
/var/displays.json	Array of 4 strings. Contains the names of all 4 output ports. Example: ["Conference Room", "Cafeteria", ...]
/var/presetnames.json	Array of 8 strings. Contains the names of all 8 presets to be set Example: ["Preset 1", "Preset 2", "week end", ...]
/var/auth.json	Array of 2 strings. Contains a single digit string representing the user level requested, and a second string representing the password. Example: ["2", "myAdminPassword"]
/var/serial.json	Array of 4 integers. Set the following serial port settings: baud rate, data bits(7 or 8), parity (0=>Even; 1=>Odd; 4=>None) ,stop bits (0=>1 stop bit; 2=>2 stop bits) Example: [9600,8,4,0]
/var/ip.json	Array of 4 integers. Set the IP address Example: [10,0,101,96] WARNING! This request will actually take effect after the DHCP request will be sent
/var/netmask.json	Array of 4 integers. Set the mask address Example: [255,255,0,0] WARNING! This request will actually take effect after the DHCP request will be sent
/var/router.json	Array of 4 integers. Set the gateway address Example: [10,0,1,1] WARNING! This request will actually take effect after the DHCP request will be sent
/var/dhcp.json	Array of a single integer. Set the DHCP state ON ([1]) or OFF ([0]) Example: [1]
/var/tempalarms.json	Array of 2 integers. Set the Min. and Max. temperature of the unit before an alarm is raised Example: [5,70]
/var/alarmflags.json	Array of 8 integers. Set the alarm configuration: beep, email, serial feedback, (the last 5 integer are not used) Example: [1,0,1,0,0,0,0,0]
var/pass_admin.json	Array of 2 strings. Set the web administrator password Example: ["old admin password", "new admin password"]
var/senddata	Hexadecimal data string to send in the request formatted as follow: Example: ["0600A010203040506070809A0"] 03: is the port number (here it is port # 3) 06: is the baud rate code

	<ul style="list-style-type: none"> ➤ For 9600 the code is 06 ➤ For 19200 the code is 08 ➤ For 38400 the code is 09 ➤ For 57600 the code is 0B ➤ For 115200 the code is 0C <p>0A: is the data length that follows (max value 0x28 bytes.)</p> <p>01A203040506070809A0: is the data in hexadecimal (0x01, 0xA2, 0x03,...)</p> <p>The response will be the data (in hexadecimal) returned if any: ["A105.....450C"]</p>
var/xsenddata	Same as the above command "var/senddata" but no response data will be returned from the device, even if there is one.
var/edidcopy_p.json	<p>Array of a single integer.</p> <p>Copy the specified predefined EDID values to all input ports.</p> <p>The different code value are:</p> <p>1 : 1080p3D 2 : 1080p 3 : 1080i 4 : 1080p 5.1 5 : 1080p 7.1</p> <p>Example to apply the predefined EDID "1080i": [3]</p>
var/edidcopy_o.json	<p>Array of 4 integers.</p> <p>Contains 4 items for the 4 sources.</p> <p>Each number is the output port (display) number that you want to copy the EDID from and copy it to the associated source. 0 means do nothing. In the example below, the EDID from the Display 2 will be copied to the source 3 and the EDID from the Display 4 will be copied to the source 1</p> <p>Example: [4,0,2,0]</p>
var/pass_user.json	<p>Array of 2 strings.</p> <p>Set the web user password</p> <p>Example: ["old admin password", "new user password"]</p>
var/reset	Reset the unit to factory settings