MGW Diamond/MGW Diamond TOUGH HTTPS API

Introduction

This document describes the **MGW Diamond/MGW Diamond TOUGH** application programming interface (**API**).
The API is used to manage and configure the appliance through an HTTPS connection.
The HTTPS API documentation shall be accessible at this location :
`https://<System_IP_Address>/doc/apidoc.html`

Connect to the API

The API can be accessed over HTTPS protocol using the below credentials:
login : `user`
password : password as set in the web interface or CLI. Default is: `1qaz!QAZ`

The authentication method is Digest.

JSON formatting

The data which are either sent with the PUT request or received with the GET request are formatted in JSON, see https://www.json.org/

This format allows a string representation of different layers of booleans, integers, strings, lists and dictionaries.

Examples of sending API requests

using cURL, from a shell

Either the `--anyauth` option or the `--digest` one can be used to specify the authentication method.

```
                  api_ipaddress=192.168.1.1
api_username='user'
api_password='1qaz!QAZ'
```

GET request

```
                    curl --anyauth -u $api_username:$api_password -k
-X GET https://$api_ipaddress/api/v1/channelsinfo
```

PUT request

```
                    curl --anyauth -u $api_username:$api_password -k
-X PUT https://$api_ipaddress/api/v1/channels/control/0 -d @- << EOF
{
  "primary": "start",
  "secondary": "stop"
}
EOF
```

using python (version 3 recommended)

There doesn’t seem to be an `anyauth` counterpart in the python `requests` module, thus we import explicitly `HTTPDigestAuth` here.

```
                  import json
import requests
from requests.auth import HTTPDigestAuth
```

```
                  api_ipaddress = '192.168.1.1'
api_username = 'user'
api_password = '1qaz!QAZ'
```

GET request

```
                    req_get = requests.get('https://%s/api/v1
/channelsinfo' % api_ipaddress,
                       auth = HTTPDigestAuth(api_username,
api_password),
                       verify = False)
status_code_get = req_get.status_code
json_formatted_data_get = req_get.text
python_formatted_data_get = json.loads(json_formatted_data_get)
```

PUT request

```
                    python_formatted_data_put = {'primary': 'start',
'secondary': 'stop'}
json_formatted_data_put = json.dumps(python_formatted_data_put)
req_put = requests.put('https://%s/api/v1/channels/control/0' %
api_ipaddress,
                       auth = HTTPDigestAuth(api_username,
api_password),
                       verify = False,
                       data = json_formatted_data_put)
status_code_put = req_put.status_code
```

Resources

/channelsinfo

Description

Retrieve channel information summary

```
GET /channelsinfo
```

Responses

HTTP Code	Description	Schema
200	Success	ChannelInformation

Example

```
                    curl --anyauth -k -X GET -u
$api_username:$api_password https://$api_ipaddress/api/v1
/channelsinfo
```

/channels

Description

Retrieve full information about encoder channels

```
GET /channels
```

Responses

HTTP Code	Description	Schema
200	Success	Channel

Example

```
                    curl --anyauth -k -X GET -u
$api_username:$api_password https://$api_ipaddress/api/v1/channels
```

/channels/<channelId>

Description

Retrieve information about specified channel

```
GET /channels/<channelId>
```

Parameters

Type	Name	Description	Schema

Type	Name	Description	Schema
Path	channelId		number

Responses

HTTP Code	Description	Schema
200	Success	Channel

Example

```
                    curl --anyauth -k -X GET -u
$api_username:$api_password https://$api_ipaddress/api/v1/channels/1
```

Description

Set channel, specified by ID inside Channel

```
PUT /channels/<channelId>
```

Parameters

Type	Name	Description	Schema
Path	channelId		number
Body	id	Channel ID	number
Body	primaryStatus	Primary Channel status	ChannelStatus
Body	secondaryStatus	Secondary Channel status	ChannelStatus
Body	name	Channel name	string
Body	source	Source settings	Source
Body	primary	Primary processing and streaming channel	Processing
Body	secondary	Secondary processing and streaming channel	Processing

Responses

HTTP Code	Description	Schema
204	Success	
400	Error	

Example

```
                    curl --anyauth -k -X PUT -u
$api_username:$api_password https://$api_ipaddress/api/v1/channels/1
```

```
-d @- << EOF
{
  "primaryStatus": "Playing",
  "secondaryStatus": "Stopped",
  "name": "1-HEVC",
  "source": {
    "input": "SDI 4",
    "format": "1280x720p @ 30",
    "forcePattern": false
  },
  "primary": {
    "bitrate": 4,
    "bitrateMax": 6,
    "rateControl": "CBR",
    "profile": "H.264 High",
    "match": false,
    "width": 1920,
    "height": 1080,
    "scanType": "Progressive",
    "framerate": 30,
    "klv1": "None",
    "level": 4.2,
    "aspectRatio": "16:9",
    "entropyMode": "CABAC",
    "gopMode": "I(2B)P",
    "gopSize": 30,
    "timestamps": true,
    "autoAdapt": true,
    "overhead": 10,
    "qpBase": 24,
    "qpMin": 16,
    "qpMax": 51,
    "delay": 120,
    "delayMax": 200,
    "pmtPID": 1240,
    "videoPID": 1241,
    "pcrPID": 1241,
    "klv1PID": 497,
    "klvSampling": false,
    "dvbMode": true,
    "rtspPort": 554,
    "target1": {
      "enable": false,
      "protocol": "UDP TS",
      "ethernetPort": "Ethernet 1",
      "ipAddress": "192.168.218.109",
      "port": 30120,
      "name": "Channel_1P2",
      "udpPacketSize": 1316,
      "TTL": 3,
      "enableTrafficShaping": false
    }
  },
  "secondary": {
    "rateControl": "CBR",
    "profile": "H.264 High",
    "match": false,
    "width": 1920,
    "height": 1080,
    "scanType": "Progressive",
    "framerate": 30,
    "klv1": "None",
    "level": 4.2,
    "aspectRatio": "16:9",
    "entropyMode": "CABAC",
    "gopMode": "I(2B)P",
    "gopSize": 30,
    "timestamps": true,
    "autoAdapt": true,
    "overhead": 10,
    "qpBase": 24,
    "qpMin": 16,
    "qpMax": 51,
    "delay": 120,
    "delayMax": 200,
    "pmtPID": 1240,
    "videoPID": 1241,
    "pcrPID": 1241,
    "klv1PID": 497,
    "klvSampling": false,
    "dvbMode": true,
    "rtspPort": 554,
    "target1": {
      "enable": false,
      "protocol": "UDP TS",
      "ethernetPort": "Ethernet 1",
      "ipAddress": "192.168.218.109",
      "port": 30120,
      "name": "Channel_2P2",
      "udpPacketSize": 1316,
      "TTL": 3,
      "enableTrafficShaping": false
    }
  }
}
EOF
```

/channels/control/<channelId>

Description

Retrieve current channel state

```
GET /channels/control/<channelId>
```

Parameters

Type	Name	Description	Schema
Path	channelId		number

Responses

HTTP Code	Description	Schema

HTTP Code	Description	Schema
200	Success	ChannelControl

Example

```
                    curl --anyauth -k -X GET -u
$api_username:$api_password https://$api_ipaddress/api/v1/channels
/control/1
```

Description

Change channel state by ID inside ChannelControl

```
PUT /channels/control/<channelId>
```

Parameters

Type	Name	Description	Schema
Path	channelId		number
Body	primary	Primary Channel State	ChannelCtrlActions
Body	secondary	Secondary Channel State	ChannelCtrlActions

Responses

HTTP Code	Description	Schema
204	Success	
400	Error	

Example

```
                    curl --anyauth -k -X PUT -u
$api_username:$api_password https://$api_ipaddress/api/v1/channels
/control/1 -d @- << EOF
{
  "primary": "start",
  "secondary": "stop"
}
EOF
```

/channels/statistics

Description

Retrieve statistics of all channels

```
GET /channels/statistics
```

Responses

HTTP Code	Description	Schema
200	Success	ChannelStatistics

Example

```
                    curl --anyauth -k -X GET -u
$api_username:$api_password https://$api_ipaddress/api/v1/channels
/statistics
```

/klv

Description

Get parameters for all KLV sources

```
                        GET /klv
```

Responses

HTTP Code	Description	Schema
200	Success	KlvSources

Example

```
                    curl --anyauth -k -X GET -u
$api_username:$api_password https://$api_ipaddress/api/v1/klv
```

Description

Set parameters for all KLV sources

```
                        PUT /klv
```

Parameters

Type	Name	Description	Schema
Body	ip1	KLV over IP Source 1	KlvoIpSource
Body	ip2	KLV over IP Source 2	KlvoIpSource
Body	ip3	KLV over IP Source 3	KlvoIpSource
Body	ip4	KLV over IP Source 4	KlvoIpSource

Type	Name	Description	Schema
Body	ip1s	KLV over IP Source 1S	KlvoIpSource
Body	ip2s	KLV over IP Source 2S	KlvoIpSource
Body	ip3s	KLV over IP Source 3S	KlvoIpSource
Body	ip4s	KLV over IP Source 4S	KlvoIpSource
Body	sdi1	KLV over SDI Source 1	KlvoSdiSource
Body	sdi2	KLV over SDI Source 2	KlvoSdiSource
Body	sdi3	KLV over SDI Source 3	KlvoSdiSource
Body	sdi4	KLV over SDI Source 4	KlvoSdiSource
Body	serial1	KLV over Serial Source 1	KlvoSerialSource
Body	serial2	KLV over Serial Source 2	KlvoSerialSource
Body	serial3	KLV over Serial Source 3	KlvoSerialSource
Body	serial4	KLV over Serial Source 4	KlvoSerialSource

Responses

HTTP Code	Description	Schema
204	Success	
400	Error	

Example

```
                    curl --anyauth -k -X PUT -u
$api_username:$api_password https://$api_ipaddress/api/v1/klv -d @-
<< EOF
{
  "ip1": {
    "ethernet": "Ethernet 1",
    "ipAddress": "192.168.218.109",
    "port": 1234,
    "encapsulation": "TS",
    "pid": 497,
    "bitrate": 200000
  },
  "sdi1": {
    "bitrate": 200000
  },
  "serial1": {
    "baudRate": "115200",
    "dataBits": "8",
    "parity": "None",
    "stopBits": "1",
    "flowControl": "None"
  }
}
```

```
EOF
```

/klv/ip/<Id>

Description

Retrieve information about specified ip KLV source

```
GET /klv/ip/<Id>
```

Parameters

Type	Name	Description	Schema
Path	Id		number

Responses

HTTP Code	Description	Schema
200	Success	KlvoIpSource

Example

```
curl --anyauth -k -X GET -u
$api_username:$api_password https://$api_ipaddress/api/v1/klv/ip/1
```

Description

Set ip KLV source, specified by ID inside KlvoIpSource

```
PUT /klv/ip/<Id>
```

Parameters

Type	Name	Description	Schema
Path	Id		number
Body	id	Source Id	number
Body	ethernet	Incoming Ethernet port	EthernetPort
Body	ipAddress	IP Address to listen to	IpAddress
Body	port	Port to listen to	number
Body	encapsulation	Type of incoming data	KlvoIpType
Body	pid	KLV PID	number
Body	bitrate	bit-rate of incoming data (b/s)	number

Responses

HTTP Code	Description	Schema
204	Success	
400	Error	

Example

```
                    curl --anyauth -k -X PUT -u
$api_username:$api_password https://$api_ipaddress/api/v1/klv/ip/1 -d
@- << EOF
{
  "ethernet": "Ethernet 1",
  "ipAddress": "192.168.218.109",
  "port": 1234,
  "encapsulation": "TS",
  "pid": 497,
  "bitrate": 200000
}
EOF
```

/klv/sdi/<Id>

Description

Retrieve information about specified sdi KLV source

```
GET /klv/sdi/<Id>
```

Parameters

Type	Name	Description	Schema
Path	Id		number

Responses

HTTP Code	Description	Schema
200	Success	KlvoSdiSource

Example

```
                    curl --anyauth -k -X GET -u
$api_username:$api_password https://$api_ipaddress/api/v1/klv/sdi/1
```

Description

Set ip KLV source, specified by ID inside KlvoSdiSource

```
PUT /klv/sdi/<Id>
```

Parameters

Type	Name	Description	Schema
Path	Id		number
Body	id	Source Id	number
Body	bitrate	bit-rate of incoming data (b/s)	number
Body	messageId	Message Id Filter	number

Responses

HTTP Code	Description	Schema
204	Success	
400	Error	

Example

```
curl --anyauth -k -X PUT -u
$api_username:$api_password https://$api_ipaddress/api/v1/klv/sdi/1
-d @- << EOF
{
  "bitrate": 200000,
  "messageId": 0
}
EOF
```

/klv/serial/<Id>

Description

Retrieve information about specified serial KLV source

```
GET /klv/serial/<Id>
```

Parameters

Type	Name	Description	Schema
Path	Id		number

Responses

HTTP Code	Description	Schema

HTTP Code	Description	Schema
200	Success	KlvoSerialSource

Example

```
                    curl --anyauth -k -X GET -u
$api_username:$api_password https://$api_ipaddress/api/v1
/klv/serial/1
```

Description

Set serial KLV source, specified by ID inside KlvoSerialSource

```
PUT /klv/serial/<Id>
```

Parameters

Type	Name	Description	Schema
Path	Id		number
Body	id	Source Id	number
Body	baudRate	baud-rate to set the serial-port to	SerialBaudRate
Body	dataBits	number of data bits in every character	SerialDataBits
Body	parity	parity mode to set the serial-port to	SerialParity
Body	stopBits	number of stop bits in every character	SerialStopBits
Body	flowControl	flow-control mode to set the serial-port to	SerialFlowControl

Responses

HTTP Code	Description	Schema
204	Success	
400	Error	

Example

```
                    curl --anyauth -k -X PUT -u
$api_username:$api_password https://$api_ipaddress/api/v1
/klv/serial/1 -d @- << EOF
{
  "baudRate": "115200",
  "dataBits": "8",
  "parity": "None",
  "stopBits": "1",
  "flowControl": "None"
}
```

```
EOF
```

/klv/statistics

Description

Retrieve statistics of all KLV inputs

```
GET /klv/statistics
```

Responses

HTTP Code	Description	Schema
200	Success	KlvStatistics

Example

```
curl --anyauth -k -X GET -u
$api_username:$api_password https://$api_ipaddress/api/v1
/klv/statistics
```

/network

Description

Retrieve device’s network configuration

```
GET /network
```

Responses

HTTP Code	Description	Schema
200	Success	Network

Example

```
curl --anyauth -k -X GET -u
$api_username:$api_password https://$api_ipaddress/api/v1/network
```

Description

Change device’s network configuration

```
PUT /network
```

Parameters

Type	Name	Description	Schema
Body	ethernet1	Parameters of Ethernet 1	NetworkAdapter
Body	ethernet2	Parameters of Ethernet 2	NetworkAdapter
Body	httpApi	Use HTTP API	boolean

Responses

HTTP Code	Description	Schema
204	Success	
400	Error	

Example

```
                    curl --anyauth -k -X PUT -u
$api_username:$api_password https://$api_ipaddress/api/v1/network -d
@- << EOF
{
  "ethernet1": {
    "ip4Type": "Static",
    "ipAddress": "192.168.1.1",
    "networkMask": "255.255.255.0",
    "defaultGateway": "192.168.1.254",
    "ip6Type": "Static",
    "ip6AutoAddress": "fe80::72b3:d5ff:fe22:bfe1",
    "ip6AutoPrefix": 64,
    "ip6ManualAddress": " ",
    "ip6ManualPrefix": 0,
    "isManagementAccess": true,
    "isPrimaryDefaultGateway": true,
    "hardwareAddress": "70:b3:d5:22:bf:e1",
    "dns": "192.168.1.5"
  },
  "ethernet2": {
    "ip4Type": "Static",
    "ipAddress": "192.168.1.2",
    "networkMask": "255.255.255.0",
    "defaultGateway": "192.168.1.254",
    "ip6Type": "Static",
    "ip6AutoAddress": "fe80::72b3:d5ff:fe22:bfe2",
    "ip6AutoPrefix": 64,
    "ip6ManualAddress": " ",
    "ip6ManualPrefix": 0,
    "isManagementAccess": true,
    "isPrimaryDefaultGateway": false,
    "hardwareAddress": "70:b3:d5:22:bf:e2",
    "dns": "192.168.1.5"
  },
  "httpApi": false
}
EOF
```

Description

/network/<Id>

Retrieve network adapter parameters

```
GET /network/<Id>
```

Parameters

Type	Name	Description	Schema
Path	Id		number

Responses

HTTP Code	Description	Schema
200	Success	NetworkAdapter

Example

```
curl --anyauth -k -X GET -u
$api_username:$api_password https://$api_ipaddress/api/v1/network/1
```

Description

Set network adapter parameters, specified by ID inside NetworkAdapter

```
PUT /network/<Id>
```

Parameters

Type	Name	Description	Schema
Path	Id		number
Body	id	ID of the adapter	number
Body	ip4Type	Method of obtaining of IPv4 address	IpV4Type
Body	ipAddress	IP Address	IpAddress
Body	networkMask	Network Mask	NetworkMask
Body	defaultGateway	Default gateway	IpAddress
Body	ip6Type	Method of obtaining of IPv6 address	IpV6Type
Body	ip6AutoAddress	IP6 automatic address	IpV6Address
Body	ip6AutoPrefix	IP6 automatic prefix	number
Body	ip6ManualAddress	IP6 manual address	IpV6Address
Body	ip6ManualPrefix	IP6 manual prefix	number

Type	Name	Description	Schema
Body	isPrimaryDefaultGateway	Default gateway of the interface is primary	boolean
Body	hardwareAddress	The MAC address of the interface	string
Body	dns	DNS address	string

Responses

HTTP Code	Description	Schema
204	Success	
400	Error	

Example

```
                    curl --anyauth -k -X PUT -u
$api_username:$api_password https://$api_ipaddress/api/v1/network/1
-d @- << EOF
{
  "ip4Type": "Static",
  "ipAddress": "192.168.1.1",
  "networkMask": "255.255.255.0",
  "defaultGateway": "192.168.1.254",
  "ip6Type": "Static",
  "ip6AutoAddress": "fe80::72b3:d5ff:fe22:bfe1",
  "ip6AutoPrefix": 64,
  "ip6ManualAddress": " ",
  "ip6ManualPrefix": 0,
  "isPrimaryDefaultGateway": true,
  "hardwareAddress": "70:b3:d5:22:bf:e1",
  "dns": "192.168.1.5"
}
EOF
```

/network/status

Description

Retrieve hardware network ports state

```
GET /network/status
```

Responses

HTTP Code	Description	Schema
200	Success	NetworkPorts

Example

```
                    curl --anyauth -k -X GET -u
$api_username:$api_password https://$api_ipaddress/api/v1/network
/status
```

/security

Description

Retrieve information about device’s security configuration

```
                        GET /security
```

Responses

HTTP Code	Description	Schema
200	Success	SystemSecurity

Example

```
                    curl --anyauth -k -X GET -u
$api_username:$api_password https://$api_ipaddress/api/v1/security
```

Description

Change device’s security configuration

```
                        PUT /security
```

Parameters

Type	Name	Description	Schema
Body	currentPassword	Current password	string
Body	newPassword	New password	string
Body	hostname	Host name	string

Responses

HTTP Code	Description	Schema
204	Success	
400	Error	

Example

```
                    curl --anyauth -k -X PUT -u
$api_username:$api_password https://$api_ipaddress/api/v1/security -d
@- << EOF
{
  "currentPassword": "Pahr3tha@t",
  "newPassword": "po+p5yeeTh",
  "hostname": "encoder25"
}
EOF
```

/banner

Description

Retrieve information about banner configuration

```
                        GET /banner
```

Responses

HTTP Code	Description	Schema
200	Success	NoticeAndConsentBanner

Example

```
                    curl --anyauth -k -X GET -u
$api_username:$api_password https://$api_ipaddress/api/v1/banner
```

Description

Change device’s banner configuration

```
                        PUT /banner
```

Parameters

Type	Name	Description	Schema
Body	enable	Display banner or not	boolean
Body	text	Banner content	LongText

Responses

HTTP Code	Description	Schema
204	Success	
400	Error	

Example

```
                    curl --anyauth -k -X PUT -u
$api_username:$api_password https://$api_ipaddress/api/v1/banner -d
@- << EOF
{
  "enable": false,
  "text": "Hello\n你好\nآلو\nЗдра́вствуйте"
}
EOF
```

/sysdatetime

Description

Retrieve information about device’s Date and Time configuration

```
GET /sysdatetime
```

Responses

HTTP Code	Description	Schema
200	Success	SystemDateTime

Example

```
                    curl --anyauth -k -X GET -u
$api_username:$api_password https://$api_ipaddress/api/v1/sysdatetime
```

Description

Change device’s dateAndTime configuration

```
PUT /sysdatetime
```

Parameters

Type	Name	Description	Schema
Body	useNtp	Use NTP server for time synchronization	boolean
Body	ntpServerAddress	IP Address of NTP server	IpAddress
Body	date	Current date	string
Body	time	Current time	string
Body	timeZone	Time Zone	string

Responses

HTTP Code	Description	Schema
204	Success	
400	Error	

Example

```
                    curl --anyauth -k -X PUT -u
$api_username:$api_password https://$api_ipaddress/api/v1/sysdatetime
-d @- << EOF
{
  "useNtp": true,
  "ntpServerAddress": "10.10.2.2",
  "date": "25.12.2016",
  "time": "01:20:34",
  "timeZone": "GMT+7"
}
EOF
```

/ports

Description

Retrieve hardware audio and video ports state

```
GET /ports
```

Responses

HTTP Code	Description	Schema
200	Success	AVPorts

Example

```
                    curl --anyauth -k -X GET -u
$api_username:$api_password https://$api_ipaddress/api/v1/ports
```

/configurations

Description

Get a list of all available configurations

```
GET /configurations
```

Responses

HTTP Code	Description	Schema
200	Success	Configuration

Example

```
                    curl --anyauth -k -X GET -u
$api_username:$api_password https://$api_ipaddress/api/v1
/configurations
```

Description

Place a new configuration

```
                        POST /configurations
```

Parameters

Type	Name	Description	Schema
Body	id	ID of the saved configuration	number
Body	name	Name of the saved configuration	string
Body	description	Short description of the saved configuration	string
Body	created	Creation date of the saved configuration	string
Body	isAutoStart	Autostart the saved configuration if true	boolean

Responses

HTTP Code	Description	Schema
201	Success	Configuration
400	Error	

Example

```
                    curl --anyauth -k -X POST -u
$api_username:$api_password https://$api_ipaddress/api/v1
/configurations -d @- << EOF
{
  "name": "Test",
  "description": "Test configuration",
  "created": "29-Aug-15 18:37:48",
  "isAutoStart": true
}
EOF
```

Description

Get the configuration with confId

/configurations/<confId>

```
GET /configurations/<confId>
```

Parameters

Type	Name	Description	Schema
Path	confId		number

Responses

HTTP Code	Description	Schema
200	Success	Configuration
404	Error	

Example

```
                    curl --anyauth -k -X GET -u
$api_username:$api_password https://$api_ipaddress/api/v1
/configurations/1
```

Description

Change configuration’s parameters (Autostart only)

```
PUT /configurations/<confId>
```

Parameters

Type	Name	Description	Schema
Path	confId		number
Body	id	ID of the saved configuration	number
Body	name	Name of the saved configuration	string
Body	description	Short description of the saved configuration	string
Body	created	Creation date of the saved configuration	string
Body	isAutoStart	Autostart the saved configuration if true	boolean

Responses

HTTP Code	Description	Schema
204	Success	

HTTP Code	Description	Schema
400	Error	
404	Not found	

Example

```
                    curl --anyauth -k -X PUT -u
$api_username:$api_password https://$api_ipaddress/api/v1
/configurations/1 -d @- << EOF
{
  "name": "Test",
  "description": "Test configuration",
  "created": "29-Aug-15 18:37:48",
  "isAutoStart": true
}
EOF
```

Description

Removes existing configuration

```
DELETE /configurations/<confId>
```

Parameters

Type	Name	Description	Schema
Path	confId		number

Responses

HTTP Code	Description	Schema
204	Success	
404	Not found	

Example

```
                    curl --anyauth -k -X DELETE -u
$api_username:$api_password https://$api_ipaddress/api/v1
/configurations/1 -d {}
```

/configurations/activate/<confId>

Description

Activate the given configuration

```
PUT /configurations/activate/<confId>
```

Parameters

Type	Name	Description	Schema
Path	confId		number

Responses

HTTP Code	Description	Schema
204	Success	
400	Error	

Example

```
                    curl --anyauth -k -X PUT -u
$api_username:$api_password https://$api_ipaddress/api/v1
/configurations/activate/1 -d {}
```

/general

Description

Retrieve General Device Information

```
GET /general
```

Responses

HTTP Code	Description	Schema
200	Success	SystemInformation

Example

```
                    curl --anyauth -k -X GET -u
$api_username:$api_password https://$api_ipaddress/api/v1/general
```

/general/restart

Description

Restart the device

```
PUT /general/restart
```

Responses

HTTP Code	Description	Schema
204	Success	
400	Error	

Example

```
                    curl --anyauth -k -X PUT -u
$api_username:$api_password https://$api_ipaddress/api/v1/general
/restart -d {}
```

/general/factoryReset

Description

Reset device to factory defaults and restart the device

```
PUT /general/factoryReset
```

Responses

HTTP Code	Description	Schema
204	Success	
400	Error	

Example

```
                    curl --anyauth -k -X PUT -u
$api_username:$api_password https://$api_ipaddress/api/v1/general
/factoryReset -d {}
```

/uptime

Description

Retrieve Date/Time Information

```
GET /uptime
```

Responses

HTTP Code	Description	Schema
200	Success	Uptime

Example

```
                    curl --anyauth -k -X GET -u
$api_username:$api_password https://$api_ipaddress/api/v1/uptime
```

/cooling

Description

Retrieve Cooling Information

```
GET /cooling
```

Responses

HTTP Code	Description	Schema
200	Success	Cooling

Example

```
                    curl --anyauth -k -X GET -u
$api_username:$api_password https://$api_ipaddress/api/v1/cooling
```

/changes

Description

Retrieve system change counters

```
GET /changes
```

Responses

HTTP Code	Description	Schema
200	Success	Changes

Example

```
                    curl --anyauth -k -X GET -u
$api_username:$api_password https://$api_ipaddress/api/v1/changes
```

Definitions

FourInteger

Array of 4 integer

SerialStatistics

Name	Description	Type
rx	Serial port Rx (KiB)	number
corrupt	Rx Corrupt Messages	number
ok	Rx OK Messages	number

KlvStatistics

Name	Description	Type
serial1	KLV over Serial Source 1 statistics	SerialStatistics
serial2	KLV over Serial Source 2 statistics	SerialStatistics
serial3	KLV over Serial Source 3 statistics	SerialStatistics
serial4	KLV over Serial Source 4 statistics	SerialStatistics

KlvoIpSource

Name	Description	Type
id	Source Id	number
ethernet	Incoming Ethernet port	EthernetPort
ipAddress	IP Address to listen to	IpAddress
port	Port to listen to	number
encapsulation	Type of incoming data	KlvoIpType
pid	KLV PID	number
bitrate	bit-rate of incoming data (b/s)	number

KlvoSdiSource

Name	Description	Type
id	Source Id	number
bitrate	bit-rate of incoming data (b/s)	number
messageId	Message Id Filter	number

KlvoSerialSource

Name	Description	Type
id	Source Id	number
baudRate	baud-rate to set the serial-port to	SerialBaudRate
dataBits	number of data bits in every character	SerialDataBits
parity	parity mode to set the serial-port to	SerialParity
stopBits	number of stop bits in every character	SerialStopBits
flowControl	flow-control mode to set the serial-port to	SerialFlowControl

KlvSources

Name	Description	Type
ip1	KLV over IP Source 1	KlvoIpSource
ip2	KLV over IP Source 2	KlvoIpSource
ip3	KLV over IP Source 3	KlvoIpSource
ip4	KLV over IP Source 4	KlvoIpSource
ip1s	KLV over IP Source 1S	KlvoIpSource
ip2s	KLV over IP Source 2S	KlvoIpSource
ip3s	KLV over IP Source 3S	KlvoIpSource
ip4s	KLV over IP Source 4S	KlvoIpSource
sdi1	KLV over SDI Source 1	KlvoSdiSource
sdi2	KLV over SDI Source 2	KlvoSdiSource
sdi3	KLV over SDI Source 3	KlvoSdiSource
sdi4	KLV over SDI Source 4	KlvoSdiSource
serial1	KLV over Serial Source 1	KlvoSerialSource
serial2	KLV over Serial Source 2	KlvoSerialSource
serial3	KLV over Serial Source 3	KlvoSerialSource
serial4	KLV over Serial Source 4	KlvoSerialSource

Range

Name	Description	Type

Name	Description	Type
startId		number
endId		number

ProMPEGsettings

Name	Description	Type
fecMode	Pro-MPEG FEC mode	FEC_Mode
matrixDimension	Pro-MPEG FEC matrix size	number
columns	Pro-MPEG FEC columns number	ProMPEG_Matrix_component
rows	Pro-MPEG FEC lines number	ProMPEG_Matrix_component
oneDportNumber	port used for Pro-MPEG 1D and 2D mode	number
twoDportNumber	additional port used for Pro-MPEG 2D mode	number

ZixiAdvanced

Name	Description	Type
password	Zixi password value	string
latency	Zixi latency value	number
enableABR	Enable/disable Zixi adaptative bitrate	boolean
maximumBitrate	Zixi maximum bitrate	number
minimumBitrate	Zixi minimum bitrate	number
enableFailover	Enable/disable Zixi failover	boolean
failoverAddress	Zixi failover address	IpAddress
failoverPort	Zixi failover port	number

ZixiStatistics

Name	Description	Type
name	Unique stream name	string
totalBitrate	Total bitrate value (kbps)	number
availableBandwidth	Available bandwidth value(kbps)	number
reconnection	Number of reconnection attempts	number

Name	Description	Type
roundTimeTrip	Round time trip (ms)	number
jitter	Zixi jitter value (ms)	number
latency	Zixi latency value (ms)	number
totalPackets	Total number of received packets	number
packetRate	Received packet rate (p/s)	number
packetLoss	Number of lost packets	number
droppedPackets	Number of recovered packets	number
recoveredPackets	Number of dropped packets	number
nonRecoveredPackets	Number of non recovered packets	number

Statistics

Name	Description	Type
zixi	zixi channel statistics	ZixiStatistics

ChannelStatistics

Name	Description	Type
name	Human readable channel name	string
primary	Primary channel statistics	Statistics
secondary	Secondary channel statistics	Statistics

ChannelsStatistics

Array of 8 ChannelStatistics

Target

Name	Description	Type
enable	Is target enabled	boolean
protocol	Streaming Protocol	StreamProtocol
ethernetPort	Outgoing ethernet port to use	EthernetPort
ipAddress	Destination IP address	IpAddress
port	Destination port	number

Name	Description	Type
name	Unique stream name	string
udpPacketSize	Max payload size	UdpPacketSize
TTL	Time to live for UDP packet	number
enableTrafficShaping	Enable/disable traffic shaping	boolean
zixiAdvanced	Zixi advanced parameters	ZixiAdvanced
proMpegAdvanced	Pro-MPEG advanced parameters	ProMPEGsettings

InputAudio

Name	Description	Type
source	Audio source	AudioSource
sampling	Audio sampling	AudioSampling
format	Audio format	AudioFormat
track	Audio track	AudioTrack
analogGain	Analog Audio Gain, in dB	number

Source

Name	Description	Type
input	Video input. Note that SDI will always take precedence over CVBS if multiple channels are using the same input connector with different SDI/CVBS settings. To have CVBS, all channels using the connector must be in CVBS.	VideoInput
format	Detected format	string
forcePattern	Always enable test pattern even if video input is present	boolean
audio1	Audio 1 source	InputAudio
cvbsHue	CVBS Hue	integer
cvbsSaturation	CVBS Saturation	integer
cvbsBrightness	CVBS Brightness	integer
cvbsContrast	CVBS Contrast	integer
cvbsNtscPedestal	CVBS NTSC Pedestal	boolean

ProcessingAudio

Name	Description	Type
enable	Process audio while encoding	boolean
codec	Audio codec	AudioCodec
mode	Audio mode	AudioType
bitrate	Audio bit-rate	AudioBitrate

Processing

Name	Description	Type
enable	Process video while encoding	boolean
videoBitrate	Video bit-rate (Mb/s) (updated on start)	number
videoBitrateMax	Max Video bit-rate (Mb/s) (updated on start)	number
bitrate	Total bit-rate (Mb/s)	number
bitrateMax	Max Total bit-rate (Mb/s)	number
codec	Stream video codec	VideoCodec
rateControl	Constant or variable Bitrate	RateControl
profile	Processing profile	ProcessingProfile
stdProfile	Standard-defined profile	StdProfile
picFormat	Picture format	PicFormat
lowLatency	Minimize end-to-end latency	boolean
match	Match output to input	boolean
width	Horizontal video frame size	number
height	Vertical video frame size	number
scanType	Scan type of video	ScanType
framerate	Encoding frame rate	number
klv1	KLV 1 type	KlvType
level	Encoding level	EncodingLevel
actualLevel	Actual Encoding level	ActualEncodingLevel
aspectRatio	Encoding aspect ratio	AspectRatio
entropyMode	Entropy coding mode	EntropyMode

Name	Description	Type
gopMode	GOP structure	GopStructure
gopSize	GOP size	number
timestamps	Timestamps	boolean
latencyMonitoring	Frame latency monitoring	boolean
autoAdapt	Automatically set overhead, QP and delays parameters	boolean
overhead	Mux rate overhead	number
qpBase	QP base	number
qpMin	QP range min	number
qpMax	QP range max	number
delay	Initial delay (ms)	number
delayMax	Maximum delay (ms)	number
pmtPID	PMT PID	number
videoPID	Video PID	number
pcrPID	PCR PID	number
audio1PID	Audio 1 PID	number
klv1PID	KLV 1 PID	number
klvSampling	KLV Sampling	boolean
dvbMode	DVB mode	boolean
rtspPort	RTSP server port	number
encryption	Encryption	boolean
aesKey	AES key	string
aesMode	AES mode	AesMode
audio1	Audio 1 processing settings	ProcessingAudio
target1	Streaming 1	Target

Channel

Name	Description	Type
id	Channel ID	number
primaryStatus	Primary Channel status	ChannelStatus
secondaryStatus	Secondary Channel status	ChannelStatus

Name	Description	Type
name	Channel name	string
source	Source settings	Source
primary	Primary processing and streaming channel	Processing
secondary	Secondary processing and streaming channel	Processing

Channels

Array of 8 Channel

ChannelControl

Name	Description	Type
primary	Primary Channel State	ChannelCtrlActions
secondary	Secondary Channel State	ChannelCtrlActions

Uptime

Name	Description	Type
dateTime	Current date/time	string
uptime	Uptime	string

Cooling

Name	Description	Type
chipTemperature	chip temperature (°C)	number
fan	Fan speed ratio (%)	number

Bitrates

Name	Description	Type
video		number
audio		number

NetworkPorts

Name	Description	Type

Name	Description	Type
ethernet1	Status of Ethernet1 port	PortStatus
ethernet2	Status of Ethernet2 port	PortStatus

AVPorts

Name	Description	Type
audio1	Status of Audio 1 port	PortStatus
audio2	Status of Audio 2 port	PortStatus
audio3	Status of Audio 3 port	PortStatus
audio4	Status of Audio 4 port	PortStatus
serial1	Status of Serial 1 port	PortStatus
serial2	Status of Serial 2 port	PortStatus
serial3	Status of Serial 3 port	PortStatus
serial4	Status of Serial 4 port	PortStatus
sdi1	Status of SDI 1 port	PortStatus
sdi2	Status of SDI 2 port	PortStatus
sdi3	Status of SDI 3 port	PortStatus
sdi4	Status of SDI 4 port	PortStatus

NetworkAdapter

Name	Description	Type
id	ID of the adapter	number
ip4Type	Method of obtaining of IPv4 address	IpV4Type
ipAddress	IP Address	IpAddress
networkMask	Network Mask	NetworkMask
defaultGateway	Default gateway	IpAddress
ip6Type	Method of obtaining of IPv6 address	IpV6Type
ip6AutoAddress	IP6 automatic address	IpV6Address
ip6AutoPrefix	IP6 automatic prefix	number
ip6ManualAddress	IP6 manual address	IpV6Address
ip6ManualPrefix	IP6 manual prefix	number

Name	Description	Type
isPrimaryDefaultGateway	Default gateway of the interface is primary	boolean
hardwareAddress	The MAC address of the interface	string
dns	DNS address	string

Network

Name	Description	Type
ethernet1	Parameters of Ethernet 1	NetworkAdapter
ethernet2	Parameters of Ethernet 2	NetworkAdapter
httpApi	Use HTTP API	boolean

SystemSecurity

Name	Description	Type
currentPassword	Current password	string
newPassword	New password	string
hostname	Host name	string

NoticeAndConsentBanner

Name	Description	Type
enable	Display banner or not	boolean
text	Banner content	LongText

SystemDateTime

Name	Description	Type
useNtp	Use NTP server for time synchronization	boolean
ntpServerAddress	IP Address of NTP server	IpAddress
date	Current date	string
time	Current time	string
timeZone	Time Zone	string

Configuration

Name	Description	Type
id	ID of the saved configuration	number
name	Name of the saved configuration	string
description	Short description of the saved configuration	string
created	Creation date of the saved configuration	string
isAutoStart	Autostart the saved configuration if true	boolean

Licenses

Name	Description	Type
channels		integer
resolution		ResolutionLicense
hevc		boolean
klv		boolean

SystemInformation

Name	Description	Type
product	Product	Product
systemSerialNumber	Device serial number	string
boardSerialNumber	Board serial number	string
hardware	Hardware version	string
software	Software version	string
licensing	Available licenses	Licenses

ChannelTargetInfo

Name	Description	Type
name	Target name	string
address	Target address	string

ProcessingInformation

Name	Description	Type
bitrate	Total bitrate	number

Name	Description	Type
resolution	Encoding format	string
audio	Audio present	boolean
status	Channel status	ChannelStatus
target1	Target 1 information	ChannelTargetInfo
klv	Is KLV used	boolean
fec	FEC error status	boolean

ChannelInformation

Name	Description	Type
name	Human readable channel name	string
source	Video input for channel	VideoInput
primary	Primary processing and streaming channel	ProcessingInformation
secondary	Secondary processing and streaming channel	ProcessingInformation

ChannelsInformation

Array of 8 ChannelInformation

Changes

Name	Description	Type
channels	Number of changes in channel list	integer
configs	Number of changes in configuration list	integer
ports	Number of changes in board ports status	integer
encoder	Number of changes in encoder parameters	FourInteger
network	Number of changes in network parameters	integer
security	Number of changes in security parameters	integer
datetime	Number of changes in datetime parameters	integer
klv	Number of changes in KLV parameters	integer
talkback	Number of changes in talkback parameters	integer
uptime	Number of changes in uptime/temperature parameters	integer
cooling	Number of changes in cooling parameters	integer

Name	Description	Type
banner	Number of changes in banner parameters	integer

Error

Name	Description	Type
message	Error message	string
property	Property caused the error	string

Enumerators

OutputResolution

Name	Value
"PAL"	0
"NTSC"	1
"720p25"	2
"720p29.97"	3
"720p30"	4
"720p50"	5
"720p59.94"	6
"720p60"	7
"1080i50"	8
"1080i59.94"	9
"1080i60"	10
"1080p23.976"	11
"1080p24"	12
"1080p25"	13
"1080p29.97"	14
"1080p30"	15
"1080p50"	16
"1080p59.94"	17
"1080p60"	18

SecondaryOutputResolution

Name	Value
"PAL"	0
"NTSC"	1
"720p25"	2
"720p29.97"	3
"720p30"	4
"720p50"	5
"720p59.94"	6
"720p60"	7
"1080i50"	8
"1080i59.94"	9
"1080i60"	10
"1080p23.976"	11
"1080p24"	12
"1080p25"	13
"1080p29.97"	14
"1080p30"	15

AspectRatio

Name	Value
"Auto"	0
"4:3"	1
"16:9"	2

TimeCode

Name	Value
"Video ES"	0
"Clock Set"	1
"Free Run"	2
"VITC Ancilliary Line"	3

VideoCodec

Name	Value

Name	Value
"H.264"	0
"HEVC"	1

ChromaSampling

Name	Value
"Monochrome"	0
"4:2:0"	1
"4:2:2"	2

ScanType

Name	Value
"Progressive"	0
"Interlaced"	1

AudioCodec

Name	Value
"MPEG"	0
"MPEG-1 Layer 1"	1
"MPEG-1 Layer 2"	2
"MPEG-1 Layer 3"	3
"AAC"	4
"AAC-LC (MPEG-2)"	5
"AAC-LC (MPEG-4)"	6
"AAC-LD"	7

StreamProtocol

Name	Value
"UDP TS"	0
"RTP TS"	1
"RTP ES (RTSP)"	2
"Zixi"	3
"Pro-MPEG"	4

FEC_Mode

Name	Value
"1-D"	0
"2-D"	1

ProMPEG_Matrix_component

Name	Value
"4"	0
"5"	1
"6"	2
"7"	3
"8"	4
"9"	5
"10"	6
"11"	7
"12"	8
"13"	9
"14"	10
"15"	11
"16"	12
"17"	13
"18"	14
"19"	15
"20"	16

ChannelStatus

Name	Value
"Playing"	0
"Stopped"	1
"Error"	2
"NotSelected"	3
"Started"	4

PortStatus

Name	Value
"Active"	0
"Inactive"	1
"Error"	2
"Unknown"	3
"Activating"	4
"Deactivating"	5

NetworkAdapterType

Name	Value
"Ethernet"	0

NetworkManagementAccess

Name	Value
"Ethernet 1 & 2"	0
"Ethernet 1"	1
"Ethernet 2"	2

EthernetPort

Name	Value
"Ethernet 1"	0
"Ethernet 2"	1

IpV4Type

Name	Value
"Static"	0
"DHCP"	1
"DHCP+Zeroconf"	2

IpV6Type

Name	Value
"Stateless"	0

Name	Value
"Stateful"	1
"Static"	2

TalkbackOption

Name	Value
"OFF"	0
"ON"	1
"ON - Push to Talk"	2

AudioLevel

Name	Value
"Line"	0
"Mic"	1

AudioType

Name	Value
"Stereo"	0
"Mono Left"	1
"Mono Right"	2

EncoderOutputResolution

Name	Value
"Same as Input"	0
"352x240p"	1
"352x288p"	2
"352x480p"	3
"352x576p"	4
"448x336p"	5
"544x480p"	6
"544x576p"	7
"640x360p"	8
"640x720p"	9

Name	Value
"704x480p"	10
"704x576p"	11
"720x480p"	12
"720x576p"	13
"960x720p"	14
"960x1080p"	15
"1280x720p"	16
"1440x1080p"	17
"1920x1080p"	18

KlvType

Name	Value
"None"	0
"SDI"	1
"IP 1"	2
"IP 2"	3
"IP 3"	4
"IP 4"	5
"IP 1S"	6
"IP 2S"	7
"IP 3S"	8
"IP 4S"	9
"Serial 1"	10
"Serial 2"	11
"Serial 3"	12
"Serial 4"	13

RateControl

Name	Value
"CBR"	0
"Capped VBR"	1

AudioBitrate

Name	Value
"16 Kb/s"	0
"24 Kb/s"	1
"32 Kb/s"	2
"48 Kb/s"	3
"56 Kb/s"	4
"64 Kb/s"	5
"96 Kb/s"	6
"112 Kb/s"	7
"128 Kb/s"	8
"160 Kb/s"	9
"192 Kb/s"	10
"256 Kb/s"	11

ProcessingProfile

Name	Value
"H.264 Baseline"	0
"H.264 Main"	1
"H.264 High"	2
"H.264 High 10"	3
"H.264 High 4:2:2"	4
"HEVC 8-Bit 4:2:0"	5
"HEVC 8-Bit 4:2:2"	6
"HEVC 10-Bit 4:2:0"	7
"HEVC 10-Bit 4:2:2"	8

StdProfile

Name	Value
"H.264 Baseline"	0
"H.264 Main"	1
"H.264 High"	2

Name	Value
"H.264 High 10"	3
"H.264 High 4:2:2"	4
"HEVC Main"	5
"HEVC Main 10bits"	6
"HEVC Main 4:2:2"	7

PicFormat

Name	Value
"4:2:0 8 bits"	0
"4:2:2 8 bits"	1
"4:2:0 10 bits"	2
"4:2:2 10 bits"	3

UdpPacketSize

Name	Value
"564"	0
"752"	1
"940"	2
"1128"	3
"1316"	4
"1472"	5

EncodingLevel

Name	Value
"Auto"	0
"1.0"	1
"1.1"	2
"1.2"	3
"1.3"	4
"2.0"	5
"2.1"	6
"2.2"	7

Name	Value
"3.0"	8
"3.1"	9
"3.2"	10
"4.0"	11
"4.1"	12
"4.2"	13
"5.0"	14
"5.1"	15
"5.2"	16
"6.0"	17
"6.1"	18
"6.2"	19

ActualEncodingLevel

Name	Value
"Undefined"	0
"1.0"	1
"1.1"	2
"1.2"	3
"1.3"	4
"2.0"	5
"2.1"	6
"2.2"	7
"3.0"	8
"3.1"	9
"3.2"	10
"4.0"	11
"4.1"	12
"4.2"	13
"5.0"	14
"5.1"	15

Name	Value
"5.2"	16
"6.0"	17
"6.1"	18
"6.2"	19

EntropyMode

Name	Value
"CABAC"	0
"CAVLC"	1

GopStructure

Name	Value
"I"	0
"IP"	1
"IBP"	2
"I(2B)P"	3
"I(3B)P"	4
"I(4B)P"	5
"I(5B)P"	6
"I(7B)P"	7

Tier

Name	Value
"Main"	0
"High"	1

ChromaBitDepth

Name	Value
"8"	0
"10"	1

OnOff

Name	Value

Name	Value
"Off"	0
"On"	1

ChannelPreset

Name	Value
"ISR Low Bandwidth"	0
"Broadcast HQ"	1
"Custom"	2

ChannelCtrlActions

Name	Value
"start"	0
"stop"	1

AudioSource

Name	Value
"SDI Embedded"	0
"Unbalanced Analog"	1

AudioSampling

Name	Value
"16 KHz"	0
"32 KHz"	1
"44.1 KHz"	2
"48 KHz"	3
"96 KHz"	4

AudioFormat

Name	Value
"PCM"	0
"AC3"	1
"DTS"	2
"AAC"	3

AudioTrack

Name	Value
"Pair 1"	0
"Pair 2"	1
"Pair 3"	2
"Pair 4"	3
"Pair 5"	4
"Pair 6"	5
"Pair 7"	6
"Pair 8"	7

VideoInput

Name	Value
"SDI 1"	0
"SDI 2"	1
"SDI 3"	2
"SDI 4"	3
"CVBS 1"	4
"CVBS 2"	5
"CVBS 3"	6
"CVBS 4"	7

KlvoIpType

Name	Value
"TS"	0
"RAW"	1

SerialBaudRate

Name	Value
"110"	0
"300"	1
"600"	2
"1200"	3

Name	Value
"2400"	4
"4800"	5
"9600"	6
"14400"	7
"19200"	8
"38400"	9
"57600"	10
"115200"	11

SerialDataBits

Name	Value
"5"	0
"6"	1
"7"	2
"8"	3

SerialParity

Name	Value
"None"	0
"Odd"	1
"Even"	2

SerialStopBits

Name	Value
"1"	0
"2"	1

SerialFlowControl

Name	Value
"None"	0
"Hardware"	1
"XON / XOFF"	2

ResolutionLicense

Name	Value
"SD"	0
"HD"	1
"4K"	2

Product

Name	Value
"MGW Diamond"	0
"MGW Diamond TOUGH"	1

AesMode

Name	Value
"128-bit"	0
"256-bit"	1

Strings

Name	Maximum length
IpAddress	45
NetworkMask	15
IpV6Address	45
LongText	2048

Upgrading

In order to upgrade the device, the following HTTP Method must be used.

Usage

```
POST  /upgrade/file
```

Example

Using curl command and default values for IP Address and password, the below commands must be applied:

- **Linux**

```
                    curl -k --anyauth --user 'user:1qaz!QAZ' -H
'Expect:'
--data-binary '@images/pkg_1.1.13.tar' 'https://192.168.1.1/upgrade
/file'
```

- **Windows**

```
                    curl -k --anyauth --user "user:1qaz!QAZ"  -H
"Expect:"
--data-binary "@pkg_1.1.13.tar" "https://192.168.1.1/upgrade/file"
```

Please note that

- The upgrade package path must be preceded by a @
- For the Linux example, the upgrade package is located in *images* subdirectory
- For the Windows example, the upgrade package is located in the current folder