Videon Streaming REST API

This document presents the REST API for Videon streaming devices. Multimedia encoder settings, decoder settings, media sources(inputs) and media destinations(outputs) can be configured through the REST API.

For specific possible values, please refer to the particular streaming device you are using. All Videon streaming devices share a common REST API, but the specific features supported will vary for each device model.

Before starting, check out our REST API Quickstart Guide. It makes getting started easy!

System

SYSTEM

GET `/v1/system` Get System Settings

Get the system-wide settings for the device.

These settings include network settings, name, id and other attributes not related to encode, decode, or media transmission.

Example URI

GET http://ip.address:2020/v1/system

Response `200` Hide

Headers

```
Content-Type: application/json
```

Body


```
{
    "name": "StreamDevice1",
    "device_roles": {
        "values": [
            "display",
            "source"
        ],
        "possible_values": [
            "strings"
        ]
    },
```


Back to top

```
    "ip_scheme": "string",
    "mac_address": "string",
    "ip_address": "ip_address",
    "static_ip_address": "ip_address",
    "static_netmask": "ip_address",
    "static_gateway": "ip_address",
    "static_dns_addresses": [
        "ip_addresses"
    ],
    "version": "string",
    "device_id": "string",
    "xml_active": {
        "value": boolean
    },
    "xml_url": "string",
    "xml_poll_interval": {
        "value": integer
    },
}
```

PUT `/v1/system` Update System Settings

Update the system-wide settings for the device.

These settings include network settings, name, id and other attributes not related to encode, decode, or media transmission.

Example URI

PUT http://ip.address:2020/v1/system

Request Hide

Headers

```
Content-Type: application/json
```

Body

```
{
    "name": "string",
    "device_roles": {
        "values": [
            "strings"
        ]
    },
    "ip_scheme": "static",
    "static_ip_address": "ip_address",
```

```
    "static_netmask": "ip_address",
    "static_gateway": "ip_address",
    "static_dns_addresses": [
        "ip_addresses"
    ],
    "xml_active": {
        "value": boolean
    },
    "xml_url": "string",
    "xml_poll_interval": {
        "value": integer
    }
}
```

Response `200` Hide

Headers

```
Content-Type: application/json
```

Components

COMPONENT LIST

GET `/v1/components` Get Component List

Get the list of components present in the device. Some Videon streaming devices have an encoder, a decoder, or both.

Example URI

GET http://ip.address:2020/v1/components

Response `200` Hide

Headers

```
Content-Type: application/json
```

Body

```
[
  {
```

```
    "name": "dec0",
    "active": boolean,
    "valid": boolean,
    "configuration_version": integer,
    "input_configuration_versions": [
      integer
    ],
    "effective_bitrate": integer
  },
  {
    "name": "enc0",
    "active": boolean,
    "valid": boolean,
    "configuration_version": integer,
    "output_configuration_versions": [
      integer,
      integer,
      integer,
      integer,
      integer,
      integer,
      integer
    ],
    "effective_bitrate": integer
  }
]
```

ENCODER CONFIGURATION

This will describe all attributes of the input video source and the encoded video output.

GET `/v1/components/enc0/configuration` Get Encoder Configuration

Get the current encoder settings.

Example URI

GET http://ip.address:2020/v1/components/enc0/configuration

Response `200` Hide

Headers

```
Content-Type: application/json
```

Body

```
{
    "video_bitrate": {
```

```
      "value": integer,
      "value_range": [
        min_integer,
        max_integer
      ]
    },
    "bitrate_mode": {
      "value": "string",
      "possible_values": [
        "strings"
      ]
    },
    "resolution": {
      "value": "string"
    },
    "video_input": {
      "value": "string",
      "possible_values": [
        "strings"
      ]
    },
    "audio_codec": {
      "value": "string",
      "possible_values": [
        "strings"
      ]
    },
    "audio_sample": {
      "value": "string",
      "possible_values": [
        "strings"
      ]
    },
    "audio_bitrate_aac": {
      "value": integer,
      "value_range": [
        min_integer,
        max_integer
      ]
    },
    "audio_mix_mode": {
      "value": "string",
      "possible_values": [
        "strings"
      ]
    },
    "latency_mode": {
      "value": "string",
      "possible_values": [
        "strings"
      ]
```

```
    },
    "keyframe_interval": {
      "interval": integer,
      "unit": "string",
      "possible_units": [
        "strings"
      ],
      "interval_range_frames": [
        min_integer,
        max_integer
      ],
      "interval_range_milliseconds": [
        min_integer,
        max_integer
      ]
    },
    "scaling_resolution": {
      "value": "string",
      "possible_values": [
        "strings"
      ]
    },
    "limit_to_30_fps": {
      "value": boolean
    },
    "h264_profile": {
      "value": "string",
      "possible_values": [
        "strings"
      ]
    },
    "video_codec": {
      "value": "string",
      "possible_values": [
        "strings"
      ]
    },
    "h265_level": {
      "value": "string"
    },
    "h265_profile": {
      "value": "string",
      "possible_values": [
        "strings"
      ]
    }
}
```

PUT `/v1/components/enc0/configuration` Update Encoder Configuration

Back to top

Update the current encoder settings.

Example URI

PUT http://ip.address:2020/v1/components/enc0/configuration

Request Hide

Headers

```
Content-Type: application/json
```

Body

```
{
    "video_bitrate": {
      "value": integer
    },
    "bitrate_mode": {
      "value": "string"
    },
    "video_input": {
      "value": "string"
    },
    "audio_codec": {
      "value": "string"
    },
    "audio_sample": {
      "value": "string"
    },
    "audio_bitrate_aac": {
      "value": integer
    },
    "audio_mix_mode": {
      "value": "string"
    },
    "latency_mode": {
      "value": "string"
    },
    "keyframe_interval": {
      "interval": integer,
      "unit": "string"
    },
    "scaling_resolution": {
      "value": "string"
    },
    "limit_to_30_fps": {
      "value": boolean
    },
```

```
    "h264_level": {
      "value": "string"
    },
    "h264_profile": {
      "value": "string"
    },
    "video_codec": {
      "value": "string"
    },
    "h265_profile": {
      "value": "string"
    }
}
```

Response `200` Hide

Headers

```
Content-Type: application/json
```

ENCODER STREAMING OUTPUTS LIST

This will return the list of all streaming outputs available on a streaming device. The encoded video produced by the encoder will be sent to each active output.

GET `/v1/components/enc0/outputs` Get Encoder Streaming Outputs List

Get the list of streaming outputs.

Example URI

GET http://ip.address:2020/v1/components/enc0/outputs

Response `200` Hide

Headers

```
Content-Type: application/json
```

Body

```
[
  {
    "name": "output0",
    "active": boolean,
    "identifier": {
      "type": "multicast",
```

```
      "multicast_ip_address": "ip_address",
      "udp_port": integer,
      "ttl": integer,
      "protocol": "string"
    }
  },
  {
    "name": "output1",
    "active": boolean,
    "identifier": {
      "type": "unicast",
      "device_id": "string",
      "ip_address": "ip_address",
      "udp_port": integer,
      "protocol": "string"
    }
  },
  {
    "name": "output2",
    "active": boolean,
    "identifier": {
      "type": "unicast",
      "device_id": "string",
      "ip_address": "ip_address",
      "udp_port": integer,
      "protocol": "string"
    }
  },
  {
    "name": "output3",
    "active": boolean,
    "identifier": {
      "type": "rtmp",
      "rtmp_status": [
        "string"
      ],
      "rtmp_service": {
        "value": "string",
        "possible_values": [
          "string"
        ]
      },
      "rtmp_service_data": "string"
    }
  },
  {
    "name": "output4",
    "active": boolean,
    "identifier": {
        "type": "file",
        "file_record": {
```

```
            "filename_base": "string",
            "drive_present": boolean,
            "drive_name": "string",
            "drive_size": integer,
            "drive_free_space": integer,
            "drive_mounted": boolean,
            "status": "string",
            "file_format: {
                "value": "string",
                "possible_values": [
                    "string"
                ]
            },
            "limit_max_file_size": {
              "active": boolean,
              "value": integer
            }
            "delete_oldest_when_full": boolean,
            "available_drives": [
                {
                    "drive_name": "string",
                    "drive_type": "string",
                    "drive_size": integer,
                    "drive_free_space": integer
                }
            ]
        },
        "ftp_upload": {
            "upload_to_ftp": boolean,
            "protocol": "string",
            "ftp_server": "string",
            "ftp_port": integer,
            "sftp_key": "string",
            "ftp_username": "string",
            "ftp_password": "string",
            "file_path": "string",
            "delete_after_upload": boolean,
            "ftp_bytes_uploaded": integer,
            "ftp_total_bytes": integer,
            "ftp_seconds_elapsed": integer,
            "ftp_files_remaining": integer,
            "status": "string"
        }
    }
}
{
    "name": "output5",
    "active": boolean,
    "identifier": {
      "type": "rtsp",
      "tcp_port": integer,
```

```
      "rtsp_stream_name": "string",
      "rtsp_demux_stream": boolean
    }
  },
  {
    "name": "output6",
    "active": boolean,
    "identifier": {
      "type": "http_pull",
      "http_streaming": {
        "protocols": [
          "strings"
        ],
        "media_container": "string",
        "segment_length": integer,
        "num_segments": integer,
      }
    }
  },
  {
    "name": "output7",
    "active": boolean,
    "identifier": {
      "type": "rtmp",
      "rtmp_status": [
        "string"
      ],
      "rtmp_service": {
        "value": "string",
        "possible_values": [
          "string"
        ]
      },
      "rtmp_service_data": "string"
    }
  },
  {
    "name": "output8",
    "active": boolean,
    "identifier": {
      "type": "rtmp",
      "rtmp_status": [
        "string"
      ],
      "rtmp_service": {
        "value": "string",
        "possible_values": [
          "string"
        ]
      },
      "rtmp_service_data": "string"
```

```
    }
  },
  {
    "name": "output9",
    "active": boolean,
    "identifier": {
      "type": "http_push",
      "http_streaming": {
        "protocols": [
          "strings"
        ],
        "media_container": "string",
        "segment_length": integer,
        "num_segments": integer,
        "push_url" : "string",
        "enable_ultra_low_latency": boolean,
        "chunk_interval": {
          "interval": integer,
          "unit": "string",
          "possible_units": [
            "strings"
          ]
        },
        "enable_synchronization": boolean,
        "dash_presentation_delay": integer
      }
    }
  },
```

ENCODER STREAMING OUTPUT

This will describe all the attributes of a single streaming output. The encoded video produced by the encoder will be sent to each active output.

GET `/v1/components/enc0/outputs/outputX` Get Encoder Streaming Output

Get the state of a single streaming output

Example URI

GET http://ip.address:2020/v1/components/enc0/outputs/outputX

Response `200` Hide

Headers

```
Content-Type: application/json
```

Body

```
{
    "name": "string",
    "active": "boolean",
    "identifier": {
        "type": "string",
        "device_id": "string",
        "multicast_ip_address": "ip_address",
        "ip_address": "ip_address",
        "udp_port": integer,
        "ttl": integer,
        "rtmp_status": [
            "string",
            "string"
        ],
        "protocol": "string",
        "rtmp_service": {
            "value": "string",
            "possible_values": [
                "string"
            ]
        },
        "rtmp_service_data": "string",
        "file_record": {
            "filename_base": "string",
            "drive_present": boolean,
            "drive_name": "string",
            "drive_size": integer,
            "drive_free_space": integer,
            "drive_mounted": boolean,
            "status": "string",
            "file_format: {
                "value": "string",
                "possible_values": [
                    "string"
                ]
            },
            "limit_max_file_size": {
              "active": boolean,
              "value": integer
            }
            "delete_oldest_when_full": boolean,
            "available_drives": [
                {
                    "drive_name": "string",
                    "drive_type": "string",
                    "drive_size": integer,
                    "drive_free_space": integer
                }
            ]
        },
```

```
        "ftp_upload": {
            "upload_to_ftp": boolean,
            "protocol": "string",
            "ftp_server": "string",
            "ftp_port": integer,
            "sftp_key": "string",
            "ftp_username": "string",
            "ftp_password": "string",
            "file_path": "string",
            "delete_after_upload": boolean,
            "ftp_bytes_uploaded": integer,
            "ftp_total_bytes": integer,
            "ftp_seconds_elapsed": integer,
            "ftp_files_remaining": integer,
            "status": "string"
        }
        "tcp_port": integer,
        "rtsp_stream_name": "string",
        "rtsp_demux_stream": boolean,
        "http_streaming": {
            "protocols": [
                "string"
            ],
            "media_container": "string",
            "segment_length": integer,
            "num_segments": integer,
            "push_url": "string",
            "enable_ultra_low_latency": boolean,
            "chunk_interval": {
                "interval": integer,
                "unit": "string",
                "possible_units": [
                    "strings"
                ]
            },
            "enable_synchronization": boolean,
            "dash_presentation_delay": integer
        }
    }
}
```

PUT `/v1/components/enc0/outputs/outputX` Update Encoder Streaming Output

Update the state of a single encoder streaming output

Example URI

PUT http://ip.address:2020/v1/components/enc0/outputs/outputX

Request Hide

Headers

```
Content-Type: application/json
```

Body

```
{
    "active": "boolean",
    "identifier": {
        "type": "string",
        "device_id": "string",
        "multicast_ip_address": "ip_address",
        "ip_address": "ip_address",
        "udp_port": integer,
        "ttl": integer,
        "protocol": "string",
        "rtmp_service": {
            "value": "string"
        },
        "rtmp_service_data": "string",
        "file_record": {
            "filename_base": "string",
            "drive_name": "string",
            "unmount_drive_now": boolean,
            "file_format: {
                "value": "string"
            },
            "limit_max_file_size": {
              "active": boolean,
              "value": integer
            }
            "delete_oldest_when_full": boolean
        },
        "ftp_upload": {
            "upload_to_ftp": boolean,
            "protocol": "string",
            "ftp_server": "string",
            "ftp_port": integer,
            "sftp_key": "string",
            "ftp_username": "string",
            "ftp_password": "string",
            "file_path": "string",
            "delete_after_upload": boolean
        }
        "tcp_port": integer,
        "rtsp_stream_name": "string",
        "rtsp_demux_stream": boolean,
        "http_streaming": {
            "media_container": "string",
```

```
            "segment_length": integer,
            "num_segments": integer,
            "push_url": "string",
            "enable_ultra_low_latency": boolean,
            "chunk_interval": {
                "interval": integer,
                "unit": "string",
            },
            "enable_synchronization": boolean,
            "dash_presentation_delay": integer
        }
    }
}
```

Response `200` Hide

Headers

```
Content-Type: application/json
```

DECODER CONFIGURATION

This will describe the settings used by the decoder to when decoding and rendering a stream.

GET `/v1/components/dec0/configuration` Get Decoder Configuration

Get the current decoder settings.

Example URI

GET http://ip.address:2020/v1/components/dec0/configuration

Response `200` Hide

Headers

```
Content-Type: application/json
```

Body

```
{
  "resolution": {
    "value": "string",
    "possible_values": [
      "strings"
    ]
```

```
  }
}
```

PUT `/v1/components/dec0/configuration` Update Decoder Configuration

Update the current decoder settings.

Example URI

PUT http://ip.address:2020/v1/components/dec0/configuration

Request Hide

Headers

```
Content-Type: application/json
```

Body

```
{
  "resolution": {
    "value": "string"
  }
}
```

Response `200` Hide

Headers

```
Content-Type: application/json
```

DECODER STREAMING INPUT

This will describe the stream input for the decoder. The decoder can only have one input, but the type can be any one of unicast, multicast, or RTMP.

GET `/v1/components/dec0/inputs` Get Decoder Streaming Input

Get the current state of the decoder streaming input.

Example URI

GET http://ip.address:2020/v1/components/dec0/inputs

Response `200` Hide

Headers

```
Content-Type: application/json
```

Body

```
{
    "name": "string",
    "active": "boolean",
    "identifier": {
        "type": "string",
        "device_id": "string",
        "multicast_ip_address": "ip_address",
        "udp_port": integer,
        "url": "string",
        "rtmp_status": [
            "string",
            "string"
        ],
        "protocol": "string"
    }
}
```

PUT `/v1/components/dec0/inputs` Update Decoder Streaming Input

Update the current state of the decoder streaming input.

Example URI

PUT http://ip.address:2020/v1/components/dec0/inputs

Request Hide

Headers

```
Content-Type: application/json
```

Body

```
{
    "active": "boolean",
    "identifier": {
        "type": "string",
        "device_id": "string",
        "multicast_ip_address": "ip_address",
```

```
        "udp_port": integer,
        "url": "string",
        "protocol": "string"
    }
}
```

Response `200` Hide

Headers

```
Content-Type: application/json
```