

REST API

Magellan™ SDN Orchestrator

SDNO Controller

Version 1.6

April 2023

Publication Information

© 2023 Imagine Communications Corp.

Proprietary and Confidential.

Imagine Communications considers this document and its contents to be proprietary and confidential. Except for making a reasonable number of copies for your own internal use, you may not reproduce this publication, or any part thereof, in any form, by any method, for any purpose, or in any language other than English without the written consent of Imagine Communications. All other uses are illegal.

This publication is designed to assist in the use of the product as it exists on the date of publication of this manual, and may not reflect the product at the current time or an unknown time in the future. This publication does not in any way warrant description accuracy or guarantee the use for the product to which it refers. Imagine Communications reserves the right, without notice to make such changes in equipment, design, specifications, components, or documentation as progress may warrant to improve the performance of the product.

Trademarks

Product names and other appropriate trademarks, e.g. D-Series™, Invenio®, PowerSmart®, Versio™ are trademarks or trade names of Imagine Communications or its subsidiaries.

Microsoft® and Windows® are registered trademarks of Microsoft Corporation. All other trademarks and trade names are the property of their respective companies.

Contact Information

Imagine Communications has office locations around the world. For domestic and international location and contact information, visit our Contact page (<https://imaginecommunications.com/contact>).

Support Contact Information

For domestic and international support contact information see:

- Support Contacts (<https://imaginecommunications.com/contact>)
- Worldwide Support e-mail (<mailto://support@imaginecommunications.com>)
- Customer Community Portal (<https://community.imaginecommunications.com/s/login/>)
- Warranty & Contract Information (<https://www.imaginecommunications.com/services/customer-care>)
- Imagine Communications Training (<https://imaginecommunications.com/services/training/>)

Contents

Revision History	6
Overview	7
REST API Structure	7
SDNO Database Structure	8
Logical Sources and Destinations	8
Devices	10
Working with Sources and Destinations	11
Source Properties	12
Request	12
Sample Response	12
Source Properties including Levels	13
Source Level Properties - Request Format	13
Source Level Properties - Sample Response	13
Source Properties including Levels and Channels	15
Source Channel Properties - Request Format	15
Source Channel Properties - Sample Response	15
Filtering Sources	23
Example: Filter Devices based on IP Address	23
Example: Filter Sources from Specific Endpoint using Endpoint Name	24
Breakaway Sources	26
Example: Filter for Breakaway Sources	26
Example: Filter for Non-Breakaway Sources	27
Retrieving (Database) Tags	28
Example: Retrieve All Tags	28
Request	28
Response	28
Example: Retrieve Specific Tags	38
Request	38
Response	38
Retrieving (Database) Salvos	39

Example: Retrieve All Salvos.....	39
Request.....	39
Response	39
Example: Retrieve Salvos by ID.....	40
Request.....	40
Response	40
Retrieving (Database) Groups.....	41
Example: Retrieving All Groups	41
Request.....	41
Response	42
Example: Retrieving Groups by ID.....	43
Request.....	43
Response	43
Retrieving (Database) Categories.....	44
Example: Retrieving all Database Categories	44
Request.....	44
Response	44
Retrieving (Database) Indexes	46
Example: Retrieving all Database Indexes.....	46
Request.....	46
Response	46
Retrieving (Database) Levels.....	48
Example: Retrieving all Database Levels	48
Request.....	48
Response	48
Working with Devices	49
Request Format	49
Sample Response.....	49
Retrieving Media Information.....	52
Media Information for IPVR Devices	52
Getting Channel Status.....	53
Getting the Multicast Address.....	54
Media Information for Non-IPVR/SDI Devices	55
Query Reference	56

Properties for Routing Control..... 57

 Using LRC Protocol..... 57

 Using X/Y Passthrough for Control 58

Revision History

Version	Date	Comments
1.0	October 2018	Draft Version
1.1	December 2019	Applies to SDNO version 2.9.8 and higher. Changes made: <ul style="list-style-type: none">• Port changed to 80 from 8935
1.2	January 2020	This update corresponds to SDNO 2.9.9. Changes made: <ul style="list-style-type: none">• Updated the Sample Response for Source Properties including Levels and Channel to show new ConnectionStatus parameter, which can have a value of either Assigned or Virtual_Embedded. See Source Channel Properties - Sample Response (on page 15)
1.3	November 2020	This update corresponds to SDNO 3.2. Changes made: <ul style="list-style-type: none">• Support for retrieving Media Information such as multicast information and whether a channel is online. See Retrieving Media Information (on page 52)
1.4	April 2021	This update corresponds to SDNO 3.4. Changes made: <ul style="list-style-type: none">• Retrieving source properties will now also return any applicable Tags. See Source Properties including Levels and Channels (on page 15)• Tags can be retrieved at the database level. See Retrieving (Database) Tags (on page 28)
1.5	April 2022	This update corresponds to SDNO 3.5. The following has been added: <ul style="list-style-type: none">• Retrieving (Database) Salvos (on page 39)• Retrieving (Database) Groups (on page 41)• Retrieving (Database) Categories (on page 44)• Retrieving (Database) Indexes (on page 46)• Retrieving (Database) Levels (on page 48)
1.6	February 2023	This update corresponds to SDNO 3.8.6. The following has been added: <ul style="list-style-type: none">• Support for identification of breakaway sources. This feature is particularly useful in case of using Magellan Touch with the SDNO. See Breakaway Sources (on page 26)• Port 80 removed from requests as it is no longer required.
1.6	April 2023	<ul style="list-style-type: none">• Minor updates to correct examples for Getting Channel Status (on page 53) and Getting the Multicast Address (on page 54).

Overview

The Magellan™ SDNO REST API provides a list of Imagine Communications defined operations for the SDNO platform, enabling third party management software to read the published SDNO database.

REST API Structure

SDNO REST API functions are structured such that you can retrieve basic details and then drill down depending on the level of granularity needed.

- Standard functions allow you to retrieve basic details (properties)
- Expand queries allow you access more details (properties) for sources/destinations
 - `$expand=levels`
 - `$expand=levels/channels`
- Filter queries allow you to filter properties based on specific values
 - `$filter={property1} eq '{value1}', {property2} eq '{value2}'`

Note that the REST API considers the last published database running on the SDNO Controller. Changes made to the SDNO system since the last publish will not be reflected.

SDNO Database Structure

At a high level, the SDNO database is composed of:

- Devices (Routers and Endpoints)
- Sources and Destinations

Logical Sources and Destinations

A Source/Destination can contain one more levels, and each level can contain one or more channels. Source/Destination properties are present in the following hierarchy:

{Source ID} or {Destination ID}

- Root Properties
 - Level Properties
 - Channel Properties

Sample Source Layout

```
{
  "Id": 204,
  "Name": "hdv_2aes_s5",
  "Alias": "",
  "LongName": "",
  "Description": "",
  "LRCIndex": 81,
  "GeneratedBreakaway": true,
  "Levels": [
    {
      "LevelId": 0,
      "LevelName": "Level 0",
      "XYIndex": "28",
      "StatusName": "hdv_2aes_s5",
      "Channels": [
        {
          "Type": "Video",
          "Router": "IPVR",
          "RouterLocation": "",
          "Endpoint": "SimSNP255",
          "EndpointLocation": "IN:Processor 1:Section 1:Program 1:Video 2",
          "EndpointData": ""
        }
      ]
    },
    {
      "LevelId": 1,
      "LevelName": "Level 1",
      "XYIndex": "58",
      "StatusName": "hdv_2aes_s5A1",

```

```
    "Channels": [
      {
        "Type": "Audio",
        "Router": "IPVR",
        "RouterLocation": "",
        "Endpoint": "SimSNP255",
        "EndpointLocation": "IN:Processor 1:Section 1:Program 1:Audio 1",
        "EndpointData": ""
      }
    ]
  },
  {
    "LevelId": 2,
    "LevelName": "Level 2",
    "XYIndex": "58",
    "StatusName": "hdv_2aes_s5A2",
    "Channels": [
      {
        "Type": "Audio",
        "Router": "IPVR",
        "RouterLocation": "",
        "Endpoint": "SimSNP255",
        "EndpointLocation": "IN:Processor 1:Section 1:Program 1:Audio 2",
        "EndpointData": ""
      }
    ]
  },
  {
    "LevelId": 3,
    "LevelName": "Level 3",
    "XYIndex": "58",
    "StatusName": "hdv_2aes_s5A3",
    "Channels": [
      {
        "Type": "Audio",
        "Router": "IPVR",
        "RouterLocation": "",
        "Endpoint": "SimSNP255",
        "EndpointLocation": "IN:Processor 1:Section 1:Program 1:Audio 1",
        "EndpointData": ""
      }
    ]
  },
  },
]
```

Note: Destination layout is similar to Source layout

Also see [Working with Sources and Destinations \(on page 11\)](#).

Devices

Devices consist of Routers and Endpoints.

- Routers include Platinum frames, IPVR Frames, Passthrough and Third-Party Frames in the SDNO system
- Endpoints are devices added as endpoints in the SDNO system; these could be third-party or Imagine

See [Working with Devices \(on page 49\)](#).

Working with Sources and Destinations

Each source/destination has basic properties at the root level, and properties for levels and channels at a lower level. Source/Destination Properties are contained in a hierarchical structure as follows:

- Root Properties
 - Levels
 - Channels

Basic/root level properties can be retrieved via a sources/destinations function.	Source Properties (on page 12)
Properties for levels and channels can be retrieved by expanding to get more details (using the \$expand query).	Source Properties including Levels (on page 13) Source Properties including Levels and Channels (on page 15)

You can also filter (using the **\$filter** query) sources/destinations to get specific information. See [Filtering Sources](#) (on page 23)

You can also filter for a list of breakaway sources. See [Breakaway Sources](#) (on page 26).

Note: Destinations work the same as sources.

Source Properties

Returns properties for all Sources in the database.

Request

```
GET {SDNO_IP}/api/system/activedb/sources
```

Sample Response

```
[
  {
    "Id": 1,
    "Name": "SW_RX_01",
    "Alias": "thSwOp01",
    "LongName": "SW_RX_01",
    "Description": "Th'Switch Feed 1",
    "LRCIndex": 1,
    "GeneratedBreakaway": false
  },
  {
    "Id": 8241,
    "Name": "MonAudNBr 1",
    "Alias": "",
    "LongName": "",
    "Description": "",
    "LRCIndex": 1559,
    "GeneratedBreakaway": false
  },
  {
    "Id": 4429,
    "Name": "SW_RX_02A1",
    "Alias": "thSwOp02A1",
    "LongName": "SW_RX_02A1",
    "Description": "Th'Switch Feed 2",
    "LRCIndex": 5,
    "GeneratedBreakaway": true
  },
  {
    "Id": 4430,
    "Name": "SW_RX_02A2",
    "Alias": "thSwOp02A2",
    "LongName": "SW_RX_02A2",
    "Description": "Th'Switch Feed 2",
    "LRCIndex": 6,
    "GeneratedBreakaway": true
  }
]
```

Source Properties including Levels

Uses an **\$expand** query to returns properties including Level properties for all Sources in the database.

Source Level Properties - Request Format

Use the basic Sources function and expand to levels using an expand query:

```
GET {SDNO_IP}/api/system/activedb/sources?$expand=levels
```

Source Level Properties - Sample Response

```
[
  {
    "Id": 1,
    "Name": "SW_RX_01",
    "Alias": "thSwOp01",
    "LongName": "SW_RX_01",
    "Description": "Th'Switch Feed 1",
    "LRCIndex": 1,
    "GeneratedBreakaway": false,
    "Levels": [
      {
        "LevelId": 0,
        "LevelName": "Level 0",
        "XYIndex": 0,
        "StatusName": "SW_RX_01",
        "Alias": "",
        "LongName": "SW_RX-01",
        "ConnectionStatus": "ASSIGNED"
      },
      {
        "LevelId": 1,
        "LevelName": "Level 1",
        "XYIndex": 0,
        "StatusName": "SW_RX_01A1",
        "Alias": "thSwOp01A1",
        "LongName": "SW_RX_01A1",
        "ConnectionStatus": "ASSIGNED"
      },
      {
        "LevelId": 2,
        "LevelName": "Level 2",
        "XYIndex": 0,
        "StatusName": "SW_RX_01A2",
        "Alias": "thSwOp01A2",
        "LongName": "SW_RX_01A2",
        "ConnectionStatus": "ASSIGNED"
      }
    ]
  }
]
```

```
    "LevelId": 9,
    "LevelName": "Level 9",
    "XYIndex": 0,
    "StatusName": "SW_RX_01D1",
    "Alias": "",
    "LongName": "SW RX-01D1",
    "ConnectionStatus": "ASSIGNED"
  }
]
},
{
  "Id": 8241,
  "Name": "MonAudNBr 1",
  "Alias": "",
  "LongName": "",
  "Description": "",
  "LRCIndex": 1559,
  "GeneratedBreakaway": false,
  "Levels": [
    {
      "LevelId": 1,
      "LevelName": "Level 1",
      "XYIndex": 4,
      "StatusName": "MonAudNBr 1",
      "Alias": "",
      "LongName": "",
      "ConnectionStatus": "ASSIGNED"
    }
  ]
},
{
  "Id": 4430,
  "Name": "SW_RX_02A2",
  "Alias": "thSwOp02A2",
  "LongName": "SW_RX_02A2",
  "Description": "Th'Switch Feed 2",
  "LRCIndex": 6,
  "GeneratedBreakaway": true,
  "Levels": [
    {
      "LevelId": 1,
      "LevelName": "Level 1",
      "XYIndex": 3,
      "StatusName": "SW_RX_02A2",
      "Alias": "thSwOp02A2",
      "LongName": "SW_RX_02A2",
      "ConnectionStatus": "ASSIGNED"
    },
    {
      "LevelId": 2,
      "LevelName": "Level 2",
      "XYIndex": 3,
      "StatusName": "SW_RX_02A2",
    }
  ]
}
```

```
[
  {
    "Alias": "thSwOp02A2",
    "LongName": "SW_RX_02A2",
    "ConnectionStatus": "ASSIGNED"
  }
]
```

Source Properties including Levels and Channels

Uses an **\$expand** query to return properties including Levels and Channels properties for all Sources in the database. Any tag definitions within the sources or logical templates will also be retrieved.

Source Channel Properties - Request Format

Use the basic Sources function and expand to Channels using an expand query:

```
GET {SDNO_IP}/api/system/activedb/sources?$expand=levels/channels
```

Source Channel Properties - Sample Response

```
[
  {
    "Id": 1,
    "Name": "SW_RX_01",
    "Alias": "thSwOp01",
    "LongName": "SW_RX_01",
    "Description": "Th'Switch Feed 1",
    "LRCIndex": 1,
    "GeneratedBreakaway": false,
    "Levels": [
      {
        "LevelId": 0,
        "LevelName": "Level 0",
        "XYIndex": 0,
        "StatusName": "SW_RX_01",
        "Alias": "",
        "LongName": "SW RX-01",
        "ConnectionStatus": "ASSIGNED",
        "Channels": [
          {
            "Id": 4424,
            "Type": "Video",
            "Router": "AristaSW",
            "RouterLocation": "",
            "Endpoint": "PROC-128A",
            "EndpointLocation": "IN:Processor 1:Section 1:Program 1:Video 2",
            "EndpointData": ""
          }
        ]
      }
    ]
  }
]
```

```
    "Tags": []
  }
]
},
{
  "LevelId": 1,
  "LevelName": "Level 1",
  "XYIndex": 0,
  "StatusName": "SW_RX_01A1",
  "Alias": "thSwOp01A1",
  "LongName": "SW_RX_01A1",
  "ConnectionStatus": "ASSIGNED",
  "Channels": [
    {
      "Id": 4425,
      "Type": "Audio",
      "Router": "AristaSW",
      "RouterLocation": "",
      "Endpoint": "PROC-128A",
      "EndpointLocation": "IN:Processor 1:Section 1:Program 1:Audio 1",
      "EndpointData": "",
      "Tags": []
    }
  ]
},
{
  "LevelId": 2,
  "LevelName": "Level 2",
  "XYIndex": 0,
  "StatusName": "SW_RX_01A2",
  "Alias": "thSwOp01A2",
  "LongName": "SW_RX_01A2",
  "ConnectionStatus": "ASSIGNED",
  "Channels": [
    {
      "Id": 4426,
      "Type": "Audio",
      "Router": "AristaSW",
      "RouterLocation": "",
      "Endpoint": "PROC-128A",
      "EndpointLocation": "IN:Processor 1:Section 1:Program 1:Audio 2",
      "EndpointData": "",
      "Tags": []
    }
  ]
},
{
  "LevelId": 9,
  "LevelName": "Level 9",
  "XYIndex": 0,
  "StatusName": "SW_RX_01D1",
  "Alias": "",
  "LongName": "SW RX-01D1",
  "ConnectionStatus": "ASSIGNED",
  "Channels": [
    {
```

```
        "Id": 4427,
        "Type": "Data",
        "Router": "AristaSW",
        "RouterLocation": "",
        "Endpoint": "PROC-128A",
        "EndpointLocation": "IN:Processor 1:Section 1:Program 1:Data 1",
        "EndpointData": "",
        "Tags": []
      }
    ]
  }
],
{
  "Id": 8241,
  "Name": "MonAudNBr 1",
  "Alias": "",
  "LongName": "",
  "Description": "",
  "LRCIndex": 1559,
  "GeneratedBreakaway": false,
  "Levels": [
    {
      "LevelId": 1,
      "LevelName": "Level 1",
      "XYIndex": 4,
      "StatusName": "MonAudNBr 1",
      "Alias": "",
      "LongName": "",
      "ConnectionStatus": "ASSIGNED",
      "Channels": [
        {
          "Id": 8242,
          "Type": "Audio",
          "Router": "AristaSW",
          "RouterLocation": "",
          "Endpoint": "PROC-128A",
          "EndpointLocation": "IN:Processor 1:Section 2:Program 1:Audio 1",
          "EndpointData": "",
          "Tags": []
        }
      ]
    }
  ]
},
{
  "Id": 8243,
  "Name": "MonAudBr 2",
  "Alias": "",
  "LongName": "",
  "Description": "",
  "LRCIndex": 1560,
  "GeneratedBreakaway": false,
  "Levels": [
    {
      "LevelId": 1,
```

```
    "LevelName": "Level 1",
    "XYIndex": 5,
    "StatusName": "MonAudBr 2",
    "Alias": "",
    "LongName": "",
    "ConnectionStatus": "ASSIGNED",
    "Channels": [
      {
        "Id": 8244,
        "Type": "Audio",
        "Router": "AristaSW",
        "RouterLocation": "",
        "Endpoint": "PROC-128A",
        "EndpointLocation": "IN:Processor 1:Section 2:Program 1:Audio 2",
        "EndpointData": "",
        "Tags": []
      }
    ]
  },
  {
    "LevelId": 2,
    "LevelName": "Level 2",
    "XYIndex": 5,
    "StatusName": "MonAudBr 2",
    "Alias": "",
    "LongName": "",
    "ConnectionStatus": "ASSIGNED",
    "Channels": [
      {
        "Id": 8249,
        "Type": "Audio",
        "Router": "AristaSW",
        "RouterLocation": "",
        "Endpoint": "PROC-128A",
        "EndpointLocation": "IN:Processor 1:Section 2:Program 1:Audio 2",
        "EndpointData": "",
        "Tags": []
      }
    ]
  }
]
},
{
  "Id": 15,
  "Name": "SW_RX_02",
  "Alias": "thSwOp02",
  "LongName": "SW_RX_02",
  "Description": "Th'Switch Feed 2",
  "LRCIndex": 4,
  "GeneratedBreakaway": false,
  "Levels": [
    {
      "LevelId": 0,
      "LevelName": "Level 0",
      "XYIndex": 1,
      "StatusName": "SW_RX_02",
```

```
"Alias": "",
"LongName": "SW_RX-02",
"ConnectionStatus": "ASSIGNED",
"Channels": [
  {
    "Id": 4428,
    "Type": "Video",
    "Router": "AristaSW",
    "RouterLocation": "",
    "Endpoint": "PROC-128A",
    "EndpointLocation": "IN:Processor 1:Section 1:Program 2:Video 2",
    "EndpointData": "",
    "Tags": []
  }
]
},
{
  "LevelId": 1,
  "LevelName": "Level 1",
  "XYIndex": 2,
  "StatusName": "SW_RX_02A1",
  "Alias": "thSwOp02A1",
  "LongName": "SW_RX_02A1",
  "ConnectionStatus": "ASSIGNED",
  "Channels": [
    {
      "Id": 4429,
      "Type": "Audio",
      "Router": "AristaSW",
      "RouterLocation": "",
      "Endpoint": "PROC-128A",
      "EndpointLocation": "IN:Processor 1:Section 1:Program 2:Audio 1",
      "EndpointData": "",
      "Tags": []
    }
  ]
},
{
  "LevelId": 2,
  "LevelName": "Level 2",
  "XYIndex": 3,
  "StatusName": "SW_RX_02A2",
  "Alias": "thSwOp02A2",
  "LongName": "SW_RX_02A2",
  "ConnectionStatus": "ASSIGNED",
  "Channels": [
    {
      "Id": 4430,
      "Type": "Audio",
      "Router": "AristaSW",
      "RouterLocation": "",
      "Endpoint": "PROC-128A",
      "EndpointLocation": "IN:Processor 1:Section 1:Program 2:Audio 2",
      "EndpointData": "",
      "Tags": []
    }
  ]
}
```

```
]
},
{
  "LevelId": 9,
  "LevelName": "Level 9",
  "XYIndex": 1,
  "StatusName": "SW_RX_02D1",
  "Alias": "",
  "LongName": "SW RX-02D1",
  "ConnectionStatus": "ASSIGNED",
  "Channels": [
    {
      "Id": 4431,
      "Type": "Data",
      "Router": "AristaSW",
      "RouterLocation": "",
      "Endpoint": "PROC-128A",
      "EndpointLocation": "IN:Processor 1:Section 1:Program 2:Data 1",
      "EndpointData": "",
      "Tags": []
    }
  ]
}
],
{
  "Id": 4429,
  "Name": "SW_RX_02A1",
  "Alias": "thSwOp02A1",
  "LongName": "SW_RX_02A1",
  "Description": "Th'Switch Feed 2",
  "LRCIndex": 5,
  "GeneratedBreakaway": true,
  "Levels": [
    {
      "LevelId": 1,
      "LevelName": "Level 1",
      "XYIndex": 2,
      "StatusName": "SW_RX_02A1",
      "Alias": "thSwOp02A1",
      "LongName": "SW_RX_02A1",
      "ConnectionStatus": "ASSIGNED",
      "Channels": [
        {
          "Id": 8245,
          "Type": "Audio",
          "Router": "AristaSW",
          "RouterLocation": "",
          "Endpoint": "PROC-128A",
          "EndpointLocation": "IN:Processor 1:Section 1:Program 2:Audio 1",
          "EndpointData": "",
          "Tags": []
        }
      ]
    }
  ]
},
{
```

```
    "LevelId": 2,
    "LevelName": "Level 2",
    "XYIndex": 2,
    "StatusName": "SW_RX_02A1",
    "Alias": "thSwOp02A1",
    "LongName": "SW_RX_02A1",
    "ConnectionStatus": "ASSIGNED",
    "Channels": [
      {
        "Id": 8246,
        "Type": "Audio",
        "Router": "AristaSW",
        "RouterLocation": "",
        "Endpoint": "PROC-128A",
        "EndpointLocation": "IN:Processor 1:Section 1:Program 2:Audio 1",
        "EndpointData": "",
        "Tags": []
      }
    ]
  },
  {
    "Id": 4430,
    "Name": "SW_RX_02A2",
    "Alias": "thSwOp02A2",
    "LongName": "SW_RX_02A2",
    "Description": "Th'Switch Feed 2",
    "LRCIndex": 6,
    "GeneratedBreakaway": true,
    "Levels": [
      {
        "LevelId": 1,
        "LevelName": "Level 1",
        "XYIndex": 3,
        "StatusName": "SW_RX_02A2",
        "Alias": "thSwOp02A2",
        "LongName": "SW_RX_02A2",
        "ConnectionStatus": "ASSIGNED",
        "Channels": [
          {
            "Id": 8247,
            "Type": "Audio",
            "Router": "AristaSW",
            "RouterLocation": "",
            "Endpoint": "PROC-128A",
            "EndpointLocation": "IN:Processor 1:Section 1:Program 2:Audio 2",
            "EndpointData": "",
            "Tags": []
          }
        ]
      }
    ]
  },
  {
    "LevelId": 2,
    "LevelName": "Level 2",
    "XYIndex": 3,
```

```
"StatusName": "SW_RX_02A2",
"Alias": "thSwOp02A2",
"LongName": "SW_RX_02A2",
"ConnectionStatus": "ASSIGNED",
"Channels": [
  {
    "Id": 8248,
    "Type": "Audio",
    "Router": "AristaSW",
    "RouterLocation": "",
    "Endpoint": "PROC-128A",
    "EndpointLocation": "IN:Processor 1:Section 1:Program 2:Audio 2",
    "EndpointData": "",
    "Tags": []
  }
]
}
```

Filtering Sources

Use a **\$filter** query to filter sources to get specific data.

Filter query syntax:

```
$filter={property1} eq '{value1}', {property2} eq '{value2}'
```

Example: Filter Devices based on IP Address

This example demonstrates how to filter devices to retrieve details for a specific endpoint based on the IP address.

Request

```
GET {SDNO_IP}/api/system/devices?$filter=comsettings/host eq  
'{IP_Address}'
```

Response

```
[  
  {  
    "Id": "cd5842da-f141-4122-8931-18155a604fb8",  
    "Name": "SimSNP255",  
    "Namespace": "IC.SNP",  
    "Type": "SNP",  
    "ComSettings": [  
      {  
        "Host": "172.25.151.255",  
        "Port": 29100,  
        "Protocol": "SEAM"  
      },  
      {  
        "Host": "172.25.152.255",  
        "Port": 29100,  
        "Protocol": "SEAM"  
      }  
    ]  
  }  
]
```

Example: Filter Sources from Specific Endpoint using Endpoint Name

This example demonstrates how to retrieve sources from a specific endpoint, based on the Endpoint Name.

Request

```
GET{SDNO_IP}/api/system/activedb/sources?$expand=levels/channels&$filter=levels/channels/endpoint eq '{Endpoint_Name}'
```

Response

```
[
  {
    "Id": 196,
    "Name": "hd_vid_s5",
    "Alias": "",
    "LongName": "",
    "Description": "",
    "LRCIndex": 77,
    "GeneratedBreakaway": true,
    "Levels": [
      {
        "LevelId": 0,
        "LevelName": "Level 0",
        "XYIndex": "24",
        "StatusName": "hd_vid_s5",
        "Channels": [
          {
            "Type": "Video",
            "Router": "IPVR",
            "RouterLocation": "",
            "Endpoint": "SimSNP255",
            "EndpointLocation": "IN:Processor 1:Section 1:Program 1:Video 2",
            "EndpointData": ""
          }
        ]
      }
    ]
  },
  {
    "Id": 228,
    "Name": "aes_s5",
    "Alias": "",
    "LongName": "",
    "Description": "",
    "LRCIndex": 93,
    "GeneratedBreakaway": true,
    "Levels": [
```

```
{
  "LevelId": 5,
  "LevelName": "Level 5",
  "XYIndex": "65",
  "StatusName": "aes_s5A1",
  "Channels": [
    {
      "Type": "Audio",
      "Router": "IPVR",
      "RouterLocation": "",
      "Endpoint": "SimSNP255",
      "EndpointLocation": "IN:Processor 1:Section 1:Program 1:Audio 1",
      "EndpointData": ""
    },
    {
      "Type": "Audio",
      "Router": "IPVR",
      "RouterLocation": "",
      "Endpoint": "SimSNP255",
      "EndpointLocation": "IN:Processor 1:Section 1:Program 1:Audio 2",
      "EndpointData": ""
    }
  ]
}
```

Breakaway Sources

Use one of the following queries to filter sources to get data on whether a source is breakaway or not.

```
$filter=GeneratedBreakaway eq true
```

```
$filter=GeneratedBreakaway eq false
```

Retrieves a list of:

- Breakaway sources in the current database, i.e. sources generated automatically as a result of the breakaway flag.
- Sources in the current database that are not breakaway.

Example: Filter for Breakaway Sources

This example demonstrates how to filter sources to retrieve breakaway sources.

Request

```
GET {SDNO_IP}/api/system/activedb/sources?$filter=GeneratedBreakaway eq true
```

Response

```
[
  {
    "Id": 4429,
    "Name": "SW_RX_02A1",
    "Alias": "thSwOp02A1",
    "LongName": "SW_RX_02A1",
    "Description": "Th'Switch Feed 2",
    "LRCIndex": 5,
    "GeneratedBreakaway": true
  },
  {
    "Id": 4430,
    "Name": "SW_RX_02A2",
    "Alias": "thSwOp02A2",
    "LongName": "SW_RX_02A2",
    "Description": "Th'Switch Feed 2",
    "LRCIndex": 6,
    "GeneratedBreakaway": true
  }
]
```

Example: Filter for Non-Breakaway Sources

This example demonstrates how to filter sources to retrieve sources that are not breakaway.

Request

```
GET {SDNO_IP}/api/system/activedb/sources?$filter=GeneratedBreakaway eq false
```

Response

```
[
  {
    "Id": 1,
    "Name": "SW_RX_01",
    "Alias": "thSwOp01",
    "LongName": "SW_RX_01",
    "Description": "Th'Switch Feed 1",
    "LRCIndex": 1,
    "GeneratedBreakaway": false
  },
  {
    "Id": 8241,
    "Name": "MonAudNBr 1",
    "Alias": "",
    "LongName": "",
    "Description": "",
    "LRCIndex": 1559,
    "GeneratedBreakaway": false
  },
  {
    "Id": 8243,
    "Name": "MonAudBr 2",
    "Alias": "",
    "LongName": "",
    "Description": "",
    "LRCIndex": 1560,
    "GeneratedBreakaway": false
  },
  {
    "Id": 15,
    "Name": "SW_RX_02",
    "Alias": "thSwOp02",
    "LongName": "SW_RX_02",
    "Description": "Th'Switch Feed 2",
    "LRCIndex": 4,
    "GeneratedBreakaway": false
  }
]
```

Retrieving (Database) Tags

Retrieves *all available* or *specific* database tags, with their properties or sets of properties (applies to Format Tags). Individual Tags can also be retrieved by name.

Example: Retrieve All Tags

This example demonstrates how to retrieve all available database tags.

Request

```
GET
{SDNO_IP}/api/system/activedb/tags?$expand=properties&$expand=sets/properties
```

Response

```
[
  {
    "Id": "ParamSet 1",
    "Name": "Default Data",
    "Group": "ParamSet",
    "Type": "Data",
    "Properties": [
      {
        "Name": "SignalFormat",
        "Value": "ST2110-40"
      },
      {
        "Name": "VPID_Code",
        "Value": ""
      },
      {
        "Name": "DID_SDID",
        "Value": ""
      },
      {
        "Name": "rtpPayloadType",
        "Value": "100"
      },
      {
        "Name": "rtpClockRate",
        "Value": "90000"
      },
      {
        "Name": "ttl",
        "Value": "64"
      }
    ]
  },
  {
```

```
"Id": "ParamSet 2",
"Name": "2110-20 720p/59",
"Group": "ParamSet",
"Type": "Video",
"Properties": [
  {
    "Name": "SignalFormat",
    "Value": "ST2110-20"
  },
  {
    "Name": "sampling",
    "Value": "YCbCr-4:2:2"
  },
  {
    "Name": "depth",
    "Value": "10"
  },
  {
    "Name": "width",
    "Value": "1280"
  },
  {
    "Name": "height",
    "Value": "720"
  },
  {
    "Name": "exactframerate",
    "Value": "60000/1001"
  },
  {
    "Name": "interlace",
    "Value": "False"
  },
  {
    "Name": "segmented",
    "Value": "False"
  },
  {
    "Name": "troff",
    "Value": ""
  },
  {
    "Name": "cmax",
    "Value": ""
  },
  {
    "Name": "colorimetry",
    "Value": "BT709"
  },
  {
    "Name": "pm",
    "Value": "2110GPM"
  },
  {
    "Name": "ssn",
    "Value": "ST2110-20:2017"
  }
]
```

```
    },
    {
      "Name": "tcs",
      "Value": ""
    },
    {
      "Name": "range",
      "Value": ""
    },
    {
      "Name": "MAXUDP",
      "Value": ""
    },
    {
      "Name": "par",
      "Value": ""
    },
    {
      "Name": "tp",
      "Value": "2110TPN"
    },
    {
      "Name": "rtpPayloadType",
      "Value": "96"
    },
    {
      "Name": "rtpClockRate",
      "Value": "90000"
    },
    {
      "Name": "ttl",
      "Value": "64"
    }
  ]
},
{
  "Id": "ParamSet 3",
  "Name": "2110-20 720p/50",
  "Group": "ParamSet",
  "Type": "Video",
  "Properties": [
    {
      "Name": "SignalFormat",
      "Value": "ST2110-20"
    },
    {
      "Name": "sampling",
      "Value": "YCbCr-4:2:2"
    },
    {
      "Name": "depth",
      "Value": "10"
    },
    {
      "Name": "width",
      "Value": "1280"
    }
  ]
}
```

```
},
{
  "Name": "height",
  "Value": "720"
},
{
  "Name": "exactframerate",
  "Value": "50"
},
{
  "Name": "interlace",
  "Value": "False"
},
{
  "Name": "segmented",
  "Value": "False"
},
{
  "Name": "troff",
  "Value": ""
},
{
  "Name": "cmax",
  "Value": ""
},
{
  "Name": "colorimetry",
  "Value": "BT709"
},
{
  "Name": "pm",
  "Value": "2110GPM"
},
{
  "Name": "ssn",
  "Value": "ST2110-20:2017"
},
{
  "Name": "tcs",
  "Value": ""
},
{
  "Name": "range",
  "Value": ""
},
{
  "Name": "MAXUDP",
  "Value": ""
},
{
  "Name": "par",
  "Value": ""
},
{
  "Name": "tp",
  "Value": "2110TPN"
```

```
    },
    {
      "Name": "rtpPayloadType",
      "Value": "96"
    },
    {
      "Name": "rtpClockRate",
      "Value": "90000"
    },
    {
      "Name": "ttl",
      "Value": "64"
    }
  ]
},
{
  "Id": "ParamSet 4",
  "Name": "Default Clock",
  "Group": "ParamSet",
  "Type": "Clock",
  "Properties": [
    {
      "Name": "mode",
      "Value": "PTP-Traceable"
    }
  ]
},
{
  "Id": "ParamSet 5",
  "Name": "2110-20 1080i/59",
  "Group": "ParamSet",
  "Type": "Video",
  "Properties": [
    {
      "Name": "SignalFormat",
      "Value": "ST2110-20"
    },
    {
      "Name": "sampling",
      "Value": "YCbCr-4:2:2"
    },
    {
      "Name": "depth",
      "Value": "10"
    },
    {
      "Name": "width",
      "Value": "1920"
    },
    {
      "Name": "height",
      "Value": "1080"
    },
    {
      "Name": "exactframerate",
      "Value": "30000/1001"
    }
  ]
}
```

```
    },  
    {  
      "Name": "interlace",  
      "Value": "True"  
    },  
    {  
      "Name": "segmented",  
      "Value": "False"  
    },  
    {  
      "Name": "troff",  
      "Value": ""  
    },  
    {  
      "Name": "cmax",  
      "Value": ""  
    },  
    {  
      "Name": "tp",  
      "Value": "2110TPN"  
    },  
    {  
      "Name": "colorimetry",  
      "Value": "BT709"  
    },  
    {  
      "Name": "pm",  
      "Value": "2110GPM"  
    },  
    {  
      "Name": "ssn",  
      "Value": "ST2110-20:2017"  
    },  
    {  
      "Name": "tcs",  
      "Value": ""  
    },  
    {  
      "Name": "range",  
      "Value": ""  
    },  
    {  
      "Name": "MAXUDP",  
      "Value": ""  
    },  
    {  
      "Name": "par",  
      "Value": ""  
    },  
    {  
      "Name": "rtpPayloadType",  
      "Value": "96"  
    },  
    {  
      "Name": "rtpClockRate",  
      "Value": "90000"  
    }
```

```
    },
    {
      "Name": "ttl",
      "Value": "64"
    }
  ]
},
{
  "Id": "ParamSet 22",
  "Name": "2110-30 8 Ch 24 Bits .125ms HCB Payload 103",
  "Group": "ParamSet",
  "Type": "Audio",
  "Properties": [
    {
      "Name": "SignalFormat",
      "Value": "ST2110-30"
    },
    {
      "Name": "bitWidth",
      "Value": "24"
    },
    {
      "Name": "numChannels",
      "Value": "8"
    },
    {
      "Name": "rtpPayloadType",
      "Value": "103"
    },
    {
      "Name": "rtpClockRate",
      "Value": "48000"
    },
    {
      "Name": "ttl",
      "Value": "64"
    },
    {
      "Name": "packetTime",
      "Value": "0.125"
    }
  ]
},
{
  "Id": "Bandwidth 1",
  "Name": "UHDVideo",
  "Group": "Bandwidth",
  "Type": "Video",
  "Properties": [
    {
      "Name": "Bandwidth",
      "Value": "11000"
    },
    {
      "Name": "Unit",
      "Value": "Mbps"
    }
  ]
}
```

```
    }
  ]
},
{
  "Id": "Bandwidth 2",
  "Name": "HDVideo",
  "Group": "Bandwidth",
  "Type": "Video",
  "Properties": [
    {
      "Name": "Bandwidth",
      "Value": "1400"
    },
    {
      "Name": "Unit",
      "Value": "Mbps"
    }
  ]
},
{
  "Id": "Bandwidth 3",
  "Name": "Audio",
  "Group": "Bandwidth",
  "Type": "Audio",
  "Properties": [
    {
      "Name": "Bandwidth",
      "Value": "4"
    },
    {
      "Name": "Unit",
      "Value": "Mbps"
    }
  ]
},
{
  "Id": "Bandwidth 4",
  "Name": "Data",
  "Group": "Bandwidth",
  "Type": "Data",
  "Properties": [
    {
      "Name": "Bandwidth",
      "Value": "5"
    },
    {
      "Name": "Unit",
      "Value": "Mbps"
    }
  ]
},
{
  "Id": "Format 1",
  "Name": "UHD",
  "Group": "Format",
  "Type": "Video",
```

```
"Sets": [
  {
    "Id": 1,
    "Name": "2160p",
    "Properties": [
      {
        "Name": "height",
        "Value": "2160"
      },
      {
        "Name": "width",
        "Value": "3840"
      },
      {
        "Name": "interlace",
        "Value": "False"
      }
    ]
  }
],
{
  "Id": "Format 2",
  "Name": "3G",
  "Group": "Format",
  "Type": "Video",
  "Sets": [
    {
      "Id": 1,
      "Name": "1080p",
      "Properties": [
        {
          "Name": "height",
          "Value": "1080"
        },
        {
          "Name": "width",
          "Value": "1920"
        },
        {
          "Name": "interlace",
          "Value": "False"
        }
      ]
    }
  ],
  {
    "Id": 2,
    "Name": "1080i",
    "Properties": [
      {
        "Name": "height",
        "Value": "1080"
      },
      {
        "Name": "width",
        "Value": "1920"
      }
    ]
  }
]
```

```
    },  
    {  
      "Name": "interlace",  
      "Value": "True"  
    }  
  ]  
},  
{  
  "Id": 3,  
  "Name": "720p",  
  "Properties": [  
    {  
      "Name": "height",  
      "Value": "720"  
    },  
    {  
      "Name": "width",  
      "Value": "1280"  
    },  
    {  
      "Name": "interlace",  
      "Value": "False"  
    }  
  ]  
},  
{  
  "Id": 4,  
  "Name": "SD",  
  "Properties": [  
    {  
      "Name": "width",  
      "Value": "720"  
    },  
    {  
      "Name": "interlace",  
      "Value": "False"  
    }  
  ]  
}  
]  
}
```

Example: Retrieve Specific Tags

This example demonstrates how to retrieve a specific database tag.

Request

```
GET
{SDNO_IP}:80/api/system/activedb/tags/'<TagName>'?$expand=properties&$expand=sets/properties
```

Response

```
{
  "Id": "Format 1",
  "Name": "UHD",
  "Group": "Format",
  "Type": "Video",
  "Sets": [
    {
      "Id": 1,
      "Name": "2160p",
      "Properties": [
        {
          "Name": "height",
          "Value": "2160"
        },
        {
          "Name": "width",
          "Value": "3840"
        },
        {
          "Name": "interlace",
          "Value": "False"
        }
      ]
    }
  ]
}
```

Retrieving (Database) Salvos

Retrieves *all available* or *specific* Salvos. Individual Salvos can also be retrieved by ID.

Salvos are retrieved with the following information:

ID	The Salvo ID
Name	The Salvo Name
Execution Option	
XY Index	The XY Index of the Salvo
LRC Index	The LRC Index of the Salvo

Example: Retrieve All Salvos

This example demonstrates how to retrieve all available Salvos.

Request

```
GET {SDNO_IP}/api/system/activedb/salvos
```

Response

```
{
  "Id": 1,
  "Name": "Set1",
  "ExecutionOption": 0,
  "XYIndex": 0,
  "LRCIndex": 1
},
{
  "Id": 2,
  "Name": "Set2",
  "ExecutionOption": 0,
  "XYIndex": 1,
  "LRCIndex": 2
},
{
  "Id": 3,
  "Name": "Set_disct",
  "ExecutionOption": 0,
  "XYIndex": 2,
  "LRCIndex": 3
},
}
```

Example: Retrieve Salvos by ID

This example demonstrates how to retrieve Salvos based on ID.

Request

```
GET {SDNO_IP}/api/system/activedb/salvos/{id}
```

Response

```
{  
  "Id": 2,  
  "Name": "Set2",  
  "ExecutionOption": 0,  
  "XYIndex": 1,  
  "LRCIndex": 2  
},
```

Retrieving (Database) Groups

Retrieves *all available* or *specific* Groups. Individual groups can be retrieved by ID.

Groups are retrieved with the following information:

ID	The Group ID
Name	The Group Name
Sources	A list of Sources in the Group
Destinations	A list of Destinations in the Group
Salvos	A list of Salvos in the Group
Subgroups	

Example: Retrieving All Groups

This example demonstrates how to retrieve all available Groups.

Request

```
GET {SDNO_IP}/api/system/activedb/groups
```

Response

```
{
  "Id": 2,
  "Name": "M7",
  "Sources": [],
  "Destinations": [
    1958,
    1960,
    1962,
    1964,
    1966,
    1968,
    1970
  ],
  "Salvos": [],
  "Subgroups": []
},
{
  "Id": 3,
  "Name": "SNP2test",
  "Sources": [],
  "Destinations": [],
  "Salvos": [],
  "Subgroups": [
    4,
    5
  ]
},
{
  "Id": 4,
  "Name": "SNP Destinations",
  "Sources": [],
  "Destinations": [
    2272,
    2291,
    2310,
    2329,
    2456,
    2475,
    2494,
    2513,
    2532,
    2534,
    2536
  ],
  "Salvos": [],
  "Subgroups": []
},
}
```

Example: Retrieving Groups by ID

This example demonstrates how to retrieve specific Groups by ID.

Request

```
GET {SDNO_IP}/api/system/activedb/groups/{id}
```

Response

```
{
  "Id": 4,
  "Name": "SNP Destinations",
  "Sources": [],
  "Destinations": [
    2272,
    2291,
    2310,
    2329,
    2456,
    2475,
    2494,
    2513,
    2532,
    2534,
    2536
  ],
  "Salvos": [],
  "Subgroups": []
}
```

Retrieving (Database) Categories

Retrieves *all available* Categories.

Categories are retrieved with the following information:

ID	The Category ID
Name	The Category Name

Example: Retrieving all Database Categories

This example demonstrates how to retrieve all available Categories.

Request

```
GET {SDNO_IP}/api/system/activedb/categories
```

Response

```
[
  {
    "Id": 1,
    "Name": "Disconnect "
  },
  {
    "Id": 9,
    "Name": "SNP1_Tx_H_C "
  },
  {
    "Id": 10,
    "Name": "SNP1_tx_H_C "
  },
  {
    "Id": 11,
    "Name": "SNP1_Tx_U_A "
  },
  {
    "Id": 12,
    "Name": "SNP1_Tx_U_B "
  },
  {
    "Id": 34,
    "Name": "SNP2_Rx_H_A "
  },
  {

```

```
[
  {
    "Id": 35,
    "Name": "SNP2_Rx_H_B "
  },
  {
    "Id": 36,
    "Name": "SNP2_Rx_U_A "
  },
  {
    "Id": 37,
    "Name": "SNP2_Rx_U_D "
  }
]
```

Retrieving (Database) Indexes

Retrieves *all available* Indexes.

Indexes are retrieved with the following information:

ID	The Index ID
Name	The Index Name

Example: Retrieving all Database Indexes

This example demonstrates how to retrieve all available Indexes.

Request

```
GET {SDNO_IP}/api/system/activedb/indexes
```

Response

```
[  
  {  
    "Id": 1,  
    "Name": "1"  
  },  
  {  
    "Id": 2,  
    "Name": "2"  
  },  
  {  
    "Id": 3,  
    "Name": "3"  
  },  
  {  
    "Id": 4,  
    "Name": "4"  
  },  
  {  
    "Id": 5,  
    "Name": "5"  
  },  
  {  
    "Id": 6,  
    "Name": "6"  
  },  
  {  
    "Id": 7,  
    "Name": "7"  
  }  
]
```

```
},  
{  
  "Id": 8,  
  "Name": "8"  
}  
]
```

Retrieving (Database) Levels

Retrieves *all* Levels in the database.

Indexes are retrieved with the following information:

ID	The Level ID
Name	The Level Name
Type	The Level Type (for example, Video or Audio)
XY Index	The XY Index of the Level
LRC Index	The LRC Index of the Level

Example: Retrieving all Database Levels

This example demonstrates how to retrieve all database Levels.

Request

```
GET {SDNO_IP}/api/system/activedb/levels
```

Response

```
{  
  "Id": 0,  
  "Name": "Video 0",  
  "Type": "Video",  
  "XYIndex": 0,  
  "LRCIndex": 1  
},  
{  
  "Id": 1,  
  "Name": "Audio 1",  
  "Type": "Audio",  
  "XYIndex": 1,  
  "LRCIndex": 2  
},  
{  
  "Id": 2,  
  "Name": "Audio 2",  
  "Type": "Audio",  
  "XYIndex": 2,  
  "LRCIndex": 3  
},  
}
```

Working with Devices

Devices include Routers and Endpoints in the SDNO system.

The Devices function returns a combined list of Endpoint Devices and Routers added to the system, including:

- Platinum Frames
- Pass-through and Third-Party Frames
- IPVR Frames
- Endpoint Devices

You can also filter (using the **\$filter** query) for more specific details, for example, devices of a particular "type".

Request Format

```
GET {SDNO_IP}/api/system/devices
```

Sample Response

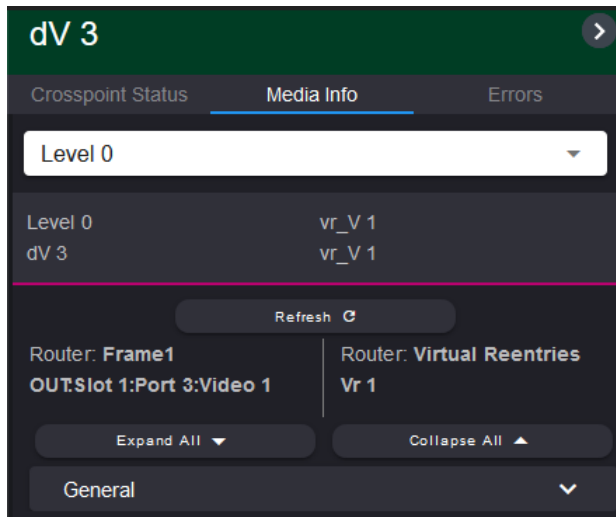
```
[
  {
    "Id": "5",
    "Name": "SimQA3",
    "Namespace": "IC.Platinum.IP3",
    "Type": "IP3 28RU",
    "ComSettings": [
      {
        "Host": "10.10.20.106",
        "Port": 52118,
        "Protocol": "IP3 PRC"
      },
      {
        "Host": "10.10.20.106",
        "Port": 4517,
        "Protocol": "CCS-P"
      },
      {
        "Host": "0.0.0.0",
        "Port": 52118,
        "Protocol": "IP3 PRC"
      },
      {
        "Host": "0.0.0.0",
```

```
        "Port": 4517,
        "Protocol": "CCS-P"
    }
]
},
{
    "Id": "5",
    "Name": "Evertz",
    "Namespace": "IC.Thirdparty.Evertz",
    "Type": "Evertz Quartz Switcher RCP",
    "ComSettings": [
        {
            "Host": "172.25.25.253",
            "Port": 52115,
            "Protocol": "Quartz Switcher RCP"
        },
        {
            "Host": "0.0.0.0",
            "Port": 52115,
            "Protocol": "Quartz Switcher RCP"
        }
    ]
},
{
    "Id": "5",
    "Name": "XYPT",
    "Namespace": "IC.Platinum.XYPassThru",
    "Type": "Imagine Communications X/Y",
    "ComSettings": [
        {
            "Host": "172.25.6.71",
            "Port": 23,
            "Protocol": "X/Y"
        },
        {
            "Host": "0.0.0.0",
            "Port": 23,
            "Protocol": "X/Y"
        }
    ]
},
{
    "Id": "4cc8612f-168d-4c66-88b3-0f6c289ce200",
    "Name": "SimEpic129",
    "Namespace": "IC.Epic-MV",
    "Type": "Epic-MV",
    "ComSettings": [
        {
            "Host": "172.25.152.129",
            "Port": 29100,
            "Protocol": "SEAM"
        }
    ],
    {
```

```
    "Host": "172.25.154.129",
    "Port": 29100,
    "Protocol": "SEAM"
  }
]
},
{
  "Id": "cd5842da-f141-4122-8931-18155a604fb8",
  "Name": "SimSNP255",
  "Namespace": "IC.SNP",
  "Type": "SNP",
  "ComSettings": [
    {
      "Host": "172.25.151.255",
      "Port": 29100,
      "Protocol": "SEAM"
    },
    {
      "Host": "172.25.152.255",
      "Port": 29100,
      "Protocol": "SEAM"
    }
  ]
},
}
]
```

Retrieving Media Information

The REST API can be used to retrieve **Media Information** for a source or destination.



See:

- [Media Information for IPVR Devices](#) (on page 52)
- [Media Information for Non-IPVR/SDI Devices](#) (on page 55)

Media Information for IPVR Devices

In case of IPVR devices, you can get:

- Channel status for sources
[Getting Channel Status](#) (on page 53)
- Multicast IP Address for destinations
[Getting the Multicast Address](#) (on page 54)

Getting Channel Status

Channel Status has 3 possible values:

- Online
- Offline
- Not Available

To get Channel status for a source or destination, use the following query:

Request

```
{SDNO_IP}/api/runtime/sources/<sourcename>.<level name>
```

```
{SDNO_IP}/api/runtime/destinations/<destinationname>.<level name>
```

Note: The source/destination name is case sensitive.

Response

```
{  
  "uri": "snp_2hdv_d 1.Level 0",  
  "channelsMediaInfo": [  
    {  
      "channelId": 118,  
      "multicastPrimary": null,  
      "multicastSecondary": null,  
      "channelStatus": "Online",  
      "metadata": {  
        "general": {  
          "signalFormat": "Video",  
          "signalStandard": "SMPTE 2110-20",  
          "program": "0"  
        }  
      }  
    }  
  ]  
}
```

Getting the Multicast Address

You can retrieve the multicast address for a source or destination.

- **Source:** For sources, see the multicast address packets are sent out on
- **Destination:** For destinations, see the multicast address the receiver is currently listening to

To get the Multicast address for a source/destination, use the following query:

Request

```
{SDNO_IP}/api/runtime/destinations/<destinationname>.<level name>
```

```
{SDNO_IP}/api/runtime/sources/<sourcename>.<level name>
```

Note: You can retrieve the multicast IP Address in case of virtuals too.

Response

```
{
  "uri": "snp_2hdv_s 1.Level 0",
  "channelsMediaInfo": [
    {
      "channelId": 94,
      "multicastPrimary": {
        "ipAddress": "239.30.1.18",
        "port": 20000
      },
      "multicastSecondary": {
        "ipAddress": "238.30.1.18",
        "port": 30000
      },
      "channelStatus": "Online",
      "metadata": {
        "general": {
          "signalFormat": "Video",
          "signalStandard": "SMPTE 2110-20",
          "program": ""
        }
      }
    }
  ]
}
```

Media Information for Non-IPVR/SDI Devices

Non-IPVR or SDI devices, like the IP3, do not have channel status and multicast address information. Instead, the video/audio information is displayed.

To retrieve information for a source or destination, use the following query:

```
{SDNO_IP}/api/runtime/sources/<sourcename>.Level<number>
```

```
{SDNO_IP}/api/runtime/destinations/<destinationname>.Level<number>
```

Channel status does not apply in case of SDI devices, and it is always be set to 0.

Figure 1: Sample SDI Source Status

```
uri: "HDebd_SRC 1.Level 0"
▼ channelsMediaInfo:
  ▼ 0:
    channelId: "1"
    multicastPrimary: null
    multicastSecondary: null
    channelStatus: 0
  ▼ metadata:
    ▼ General:
      signalFormat: "VIDEO"
```

Query Reference

Queries can be used in source/destination as well as router/endpoint functions.

Query Type	Applies to	Query Syntax	Description
Expand	Properties	<code>\$expand=levels</code> <code>\$expand=levels/channels</code>	Expands the list to include properties in the Levels and Levels/Channels sub folders
Filter	Data	<code>\$filter={property1} eq '{value1}', {property2} eq '{value2}'</code>	Filters data using the provided (comma separated) property values

Properties for Routing Control

Data retrieved via REST API functions can be used by Imagine and third-party clients (that interact with the SDNO system) to perform functions such as TAKES.

Router Control on the SDNO can be done in two ways:

- LRC
- X/Y Passthrough

Using LRC Protocol

Logical Router Control Protocol (LRC) allows control and monitoring of routers and router crosspoints based on the names and/or IDs assigned in the Logical Database.

LRC Control Using Name

Use the highlighted properties:

```
{
  "Id": 0,
  "Name": "snp1",
  "Alias": "",
  "LongName": "",
  "Description": "",
  "LRCIndex": 1,
  "Levels": [
    {
      "LevelId": 6,
      "LevelName": "Level 6",
      "XYIndex": "0",
      "StatusName": "hd_emb_s1",
      "Channels": [
        {
          "Type": "Video",
          "Router": "SimQA3",
          "RouterLocation": "IN:Slot 53:Port 1:Video 1",
          "Endpoint": "",
          "EndpointLocation": "",
          "EndpointData": ""
        }
      ]
    }
  ]
},
```

LRC Control using LRC Index:

Use the highlighted properties:

<pre> { "Id": 0, "Name": "snp1", "Alias": "", "LongName": "", "Description": "", "LRCIndex": 1, "Levels": [{ "LevelId": 6, "LevelName": "Level 6", "XYIndex": "0", "StatusName": "hd_emb_s1", "Channels": [{ "Type": "Video", "Router": "SimQA3", "RouterLocation": "IN:Slot 53:Port 1:Video 1", "Endpoint": "", "EndpointLocation": "", "EndpointData": "" }] }] }, </pre>	<pre> { "Id": 0, "Name": "Level 0", "XYIndex": 0, "LRCIndex": 1 }, { "Id": 1, "Name": "Level 1", "XYIndex": 1, "LRCIndex": 2 }, { "Id": 2, "Name": "Level 2", "XYIndex": 2, "LRCIndex": 3 }, }, </pre>
<p>LRC Index in the Source Details</p>	<p>LRC Index in the Level DB</p>

Using X/Y Passthrough for Control

X/Y Control

<pre> { "Id": 0, "Name": "snp1", "Alias": "", "LongName": "", "Description": "", "LRCIndex": 1, "Levels": [{ "LevelId": 6, "LevelName": "Level 6", "XYIndex": "0", "StatusName": "hd_emb_s1", "Channels": [{ "Type": "Video", "Router": "SimQA3", "RouterLocation": "IN:Slot 53:Port 1:Video 1", "Endpoint": "", "EndpointLocation": "", "EndpointData": "" }] }] }, </pre>	<pre> { "Id": 0, "Name": "Level 0", "XYIndex": 0, "LRCIndex": 1 }, { "Id": 1, "Name": "Level 1", "XYIndex": 1, "LRCIndex": 2 }, { "Id": 2, "Name": "Level 2", "XYIndex": 2, "LRCIndex": 3 }, }, </pre>
<p>X/Y Index in the Source Details</p>	<p>X/Y Index in the Level DB</p>