REST API for Scalable Tape Libraries

Table of Contents

Chapter 1. Overview

A simple API to manage the Scalable Tape Libraries remotely over a HTTPS interface. This is requested and needed for manufacturing and for automated test / monitoring systems.

1.1. Version information

Version : 1.1.22

1.2. URI scheme

Schemes : HTTPS

1.3. Tags

- System : System requests
- Deprecated : Deprecated Commands
- Library Information : Library Status / Information
- Library Operations : Library Operations
- Library Configuration : Library Configuration
- Library Test : Library Tests
- Drive Configuration : Drive Configuration
- V1-Authentication : Authentication request
- V1-Accessor : Accessor requests
- V1-Drives : Drives requests
- V1-Ethernet : Ethernet Ports requests
- V1-Events : Events requests
- V1-Library : Library requests
- V1-LogicalLibrary : Logical Library Information
- V1-Reports : Reports requests
- V1-System : System requests

1.4. General REST API Info

Name	Value
Protocol	HTTPS
Port	3031
Authorization	Bearer token

Name	Value
Base Path	/
Certificate	Self-Signed
Example URL	https://libraryip:3031/v1/apiversion

1.5. How To Use the REST API

The REST API workflow is as follows:

1. Login to the system with the **POST** command URL https://libraryip:3031/v1/login, where the username and password must be sent as content in JSON format **(application/json)**.
   Example Content:
   {
   "username" : "administrator",
   "password" : "password"
   }
2. If the login was successful a Bearer token will be returned like: "Bearer eyJ0eXAiOiJKV1QiLCJhb....zRcVud3c"
3. The returned Bearer token must be added as the "Authorization" header value to the REST calls which have a security check.
   Authorization: Bearer token

NOTE: If the token is invalid or the login was invalid a 401 HTTP error will be returned.

CURL Examples:

curl -X **GET** "https://libraryip:3031/v1/accessor" -H "accept: application/json" -H "Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhb....zRcVud3c" -k

1.6. Synonyms

To generalize the document and to support different wording in different environments the following synonyms are used in this specification

Name	Synonym	Description
I/O Station	Mailslot	The removable media slots in a magazine
Logical Library	Partition	The virtualization of the physical library into multiple logical partitions, which shows up for a host a tape library
Tape, Cartridge	Media	The data tape

Name	Synonym	Description
VolSer	Barcode	The barcode label which is located on the back of the tape media

Chapter 2. Resources

2.1. System

System requests

2.1.1. Get the current REST API version

```
GET /rest/apiversion
```

Description

The REST API version used in this software release

Responses

HTTP Code	Description	Schema
200	OK	Response 200
500	Error occurred at operation	Error

Response 200

Name	Schema
version *required*	string

Example HTTP response

Response 200

```
{
  "version" : "1.1.10"
}
```

2.1.2. Login Request

```
POST /rest/login
```

Description

Required Initial request to get authenticated against the tape library. The user roles/names of the library are used. This API command returns after a successful authentication an session based Bearer token, which has to be attached in the "Authorization" header of every following API command during a session.

Example using curl: " curl -X GET 'https://<library-ip>:3031/rest/library/baseinfo' -H 'accept: application/json' -H 'Authorization: Bearer token' "

Parameters

Type	Name	Description	Schema
Body	**user** *required*	the user/password which should be used to log in.	user

user

Name	Description	Schema
username *required*	**Example** : `"administrator"`	string
password *required*	**Example** : `"password"`	string
service_password *optional*	service password in case the administrator password is additionally required **Example** : `"servicepwd"`	string
change_password *optional*	If account needs a password change after valid login credentials provided this param needs to be set with the new password	string

Responses

HTTP Code	Description	Schema
201	Login request was successful and session token is generated	Response 201
401	Access denied - not authenticated	No Content
500	Error occurred at operation	Error

Response 201

Name	Schema
token *required*	string

Consumes

- `application/json`

Example HTTP response

Response 201


```
{
  "status" : "ok",
  "token" : "Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpYXQiOjEwMDg5OSwiZXhwIjoxMDgwOTksIlVST0xFIjoyL
CJqaXQiOiI0NmEzOTBiZDcwMTA5Y2ViYjk0MzExNTQ2YjkxYjI3ZXQyM2Y1Y2hhYmE5bnNpdTQ3bnU3cXI1a3Q
zMDAwMThhMjM2ZDQ5MDYuNTg2OTc4MTcifQ.DrCpCaX0jMwREFRYmSHazGY7wOs6QfGzRcVud3cDjA0"
}
```


2.2. Deprecated

Deprecated Commands

2.2.1. Get drive information

```
GET /rest/drive/information
```

Description

Deprecated: As of release 1.5.0.0-A00 this command should no longer be used. Please use /v1/drive/information instead.

Returns the information about drives like Product Id, Model, Serialno and other useful information.

Responses

HTTP Code	Description	Schema
200	OK	< DriveInfo > array

Security

Type	Name
apiKey	UserSecurity

Example HTTP response

Response 200


```
{
  "LogicalNumber" : "1",
  "PhysicalNumber" : "10",
  "Module" : "1",
  "Partition" : "0",
  "Generation" : "8",
  "Cartridge" : "FALSE",
  "Vendor" : "IBM",
  "Product" : "ULT3580-HH8",
  "FWRevision" : "J28F",
  "SerialNumber" : "000000005B",
  "WWNodeName" : "",
  "Interface" : "FC",
  "MFGSerialNumber" : "1013000188",
  "ErrorState" : "TRUE",
  "Power" : "FALSE",
  "Presence" : "TRUE",
  "ADTMode" : "IADT"
}
```


2.2.2. Inventory of physical library

```
GET /rest/library/inventory
```

Description

Deprecated: As of release 1.5.0.0-A00 this command should no longer be used. Please use /v1/library/inventory instead.

Returns the inventory of the physical library including slots, drives and mailslots. It also returns the corresponding metadata which belongs to the different types of the elements

Responses

HTTP Code	Description	Schema
200	OK	Inventory
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficent rights	No Content
500	Error occurred at operation	Error

Produces

- `application/json`

Security

Type	Name
apiKey	**UserSecurity**

Example HTTP response

Response 200


```
{
  "Slots" : [ {
    "PhysicalNumber" : "1",
    "LogicalNumber" : "1.1",
    "Module" : "1",
    "Partition" : "1",
    "Mailslot" : "FALSE",
    "Cartridge" : "FALSE",
    "CartridgeType" : "N/A",
    "CartridgeSubType" : 0,
    "Access" : "TRUE",
    "Blocked" : "TRUE"
  }, {
    "PhysicalNumber" : "2",
    "LogicalNumber" : "1.2",
    "Module" : "1",
    "Partition" : "1",
    "Mailslot" : "FALSE",
    "Cartridge" : "FALSE",
    "CartridgeType" : "N/A",
    "CartridgeSubType" : 0,
    "Access" : "TRUE",
    "Blocked" : "FALSE"
```

```
  }, {
    "PhysicalNumber" : "3",
    "LogicalNumber" : "1.3",
    "Module" : "1",
    "Partition" : "1",
    "Mailslot" : "FALSE",
    "Cartridge" : "TRUE",
    "Barcode" : "TT0013L4",
    "CartridgeType" : "Data",
    "CartridgeSubType" : 0,
    "CartridgeGeneration" : "4",
    "CartridgeEncrypted" : "FALSE",
    "Access" : "TRUE",
    "Blocked" : "FALSE"
  }, { }, {
    "PhysicalNumber" : "280",
    "LogicalNumber" : "7.40",
    "Module" : "7",
    "Partition" : "0",
    "Mailslot" : "TRUE",
    "Cartridge" : "FALSE",
    "CartridgeType" : "N/A",
    "CartridgeSubType" : 0,
    "Access" : "TRUE",
    "Blocked" : "FALSE"
  } ],
  "Drives" : [ {
    "PhysicalNumber" : "10",
    "LogicalNumber" : "1",
    "Partition" : "0",
    "Vendor" : "VENDOR  ",
    "Product" : "Ultrium 7-SCSI  ",
    "FWRevision" : "ABCD",
    "SerialNumber" : 1234567890
  }, {
    "PhysicalNumber" : "11",
    "LogicalNumber" : "2",
    "Partition" : "0",
    "Vendor" : "VENDOR  ",
    "Product" : "Ultrium 7-SCSI  ",
    "FWRevision" : "ABCD",
    "SerialNumber" : 1234567891
  }, {
    "PhysicalNumber" : "12",
    "LogicalNumber" : "3",
    "Partition" : "0",
    "Vendor" : "VENDOR  ",
    "Product" : "Ultrium 7-SCSI  ",
    "FWRevision" : "ABCD",
    "SerialNumber" : 1234567892
  }, {
```

```
    "PhysicalNumber" : "20",
    "LogicalNumber" : "4",
    "Partition" : "0",
    "Vendor" : "",
    "Product" : "",
    "FWRevision" : "",
    "SerialNumber" : ""
  } ]
}
```

2.2.3. List of media

```
GET /rest/library/mediainfo
```

Description

Deprecated: As of release 1.5.0.0-A00 this command should no longer be used. Please use /v1/library/mediainfo instead.

Returns the list of media which are located in the library including detailed information about them

Responses

HTTP Code	Description	Schema
200	OK	< MediaInfoData > array
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficent rights	No Content
500	Error occurred at operation	Error

Produces

- `application/json`

Security

Type	Name
apiKey	**UserSecurity**

Example HTTP response

Response 200

```
[ {
  "Barcode" : "TC070ML5",
  "LocationType" : "SLOT",
  "LogicalNumber" : "2.34",
  "PhysicalNumber" : "74",
  "Cleaning" : "FALSE",
  "Partition" : "1",
  "Generation" : "5",
  "SubType" : "0",
  "Protection" : "FALSE",
  "Encryption" : "FALSE",
  "NoLoads" : "148",
  "MBRead" : "153",
  "MBReadLoad" : "150",
  "MBWritten" : "226",
  "MBWrittenLoad" : "226"
}, {
  "Barcode" : "000113L5",
  "LocationType" : "SLOT",
  "LogicalNumber" : "3.4",
  "PhysicalNumber" : "84",
  "Cleaning" : "FALSE",
  "Partition" : "1",
  "Generation" : "5",
  "SubType" : "0",
  "Protection" : "FALSE",
  "Encryption" : "FALSE",
  "NoLoads" : "11",
  "MBRead" : "0",
  "MBReadLoad" : "0",
  "MBWritten" : "14",
  "MBWrittenLoad" : "14"
}, {
  "Barcode" : "TC084ML5",
  "LocationType" : "SLOT",
  "LogicalNumber" : "3.28",
  "PhysicalNumber" : "108",
  "Cleaning" : "FALSE",
  "Partition" : "2",
  "Generation" : "5",
  "SubType" : "0",
  "Protection" : "TRUE",
  "Encryption" : "FALSE",
  "NoLoads" : "64",
  "MBRead" : "1643",
  "MBReadLoad" : "1603",
  "MBWritten" : "943",
  "MBWrittenLoad" : "943"
} ]
```

2.2.4. Create partitions in simple mode

```
POST /rest/partition/createsimple
```

Description

Deprecated: As of release 1.5.0.0-A00 this command should no longer be used. Please use /v1/logicalLibrary/createsimple instead.

Creates the number of partitions which are requested in the parameters. This commands corresponds to the Basic Wizard in RMI to create partitions. The maximum number of partitions which can be created are limited by the number of installed drives.

Parameters

Type	Name	Description	Schema
Body	**partitionParameters** *required*	the parameters to create the partitions	partitionParameters

partitionParameters

Name	Description	Schema
numPartitions *required*	The number of partitions to create **Minimum value** : 1 **Maximum value** : 21 **Example** : 1	integer
barcodeLength *optional*	**Minimum value** : 6 **Maximum value** : 16 **Example** : 8	integer
barcodeAlignment *optional*	Alignment of the barcode for SCSI element status. Can be "left" or "right" **Default** : "left"	enum (left, right)
autoClean *optional*	Auto Cleaning for this partition. 'TRUE' for enabling 'FALSE' to disable it. **Default** : "FALSE"	enum (TRUE, FALSE)
sequentialMode *optional*	Enable Sequential Mode for this partition. 'TRUE' for enabling 'FALSE' to disable it. **Default** : "FALSE"	enum (TRUE, FALSE)

Name	Description	Schema
sequentialModeLoop *optional*	Restarting sequence when finished. Requires sequentialMode to be 'TRUE' **Default** : `"FALSE"`	enum (TRUE, FALSE)
sequentialModeAutoload *optional*	Load first cartridge automatically in sequential mode. Requires sequentialMode to be 'TRUE' **Default** : `"FALSE"`	enum (TRUE, FALSE)

Responses

HTTP Code	Description	Schema
200	OK	No Content
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficent rights	No Content
500	Error occurred at operation	Error

Consumes

- `application/json`

Security

Type	Name
apiKey	**AdminSecurity**

2.2.5. Get partition (Logical Library) information

```
GET /rest/partition/information
```

Description

Deprecated: As of release 1.5.0.0-A00 this command should no longer be used. Please use /v1/logicalLibrary/information instead.

The command returns the list of partitions including partition details.

Responses

HTTP Code	Description	Schema
200	OK	< PartitionInfo > array
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficent rights	No Content
500	Error occurred at operation	Error

Produces

- `application/json`

Security

Type	Name
apiKey	**UserSecurity**

Example HTTP response

Response 200


```
[ {
  "PartitionNumber" : "1",
  "Name" : "New Partition_1",
  "SerialNumber" : "DE00000000_LL01",
  "NumSlots" : "87",
  "NumIOSlots" : "0",
  "NumDrives" : "1",
  "LunMasterDrive" : "1",
  "LunMasterDrivePhys" : "10",
  "LunMasterDriveArr" : [ "1" ],
  "LunMasterDrivePhysArr" : [ "10" ],
  "EncryptionMode" : "ISV",
  "BarcodeAlign" : "left",
  "BarcodeLength" : "8",
  "AutoClean" : "TRUE",
  "WWNode" : "5001000000000071",
  "Micw" : "FALSE"
}, {
  "PartitionNumber" : "2",
  "Name" : "New Partition_2",
  "SerialNumber" : "DE00000000_LL02",
```

```
  "NumSlots" : "90",
  "NumIOSlots" : "5",
  "NumDrives" : "1",
  "LunMasterDrive" : "2",
  "LunMasterDrivePhys" : "11",
  "LunMasterDriveArr" : [ "2" ],
  "LunMasterDrivePhysArr" : [ "11" ],
  "EncryptionMode" : "ISV",
  "BarcodeAlign" : "left",
  "BarcodeLength" : "8",
  "AutoClean" : "TRUE",
  "WWNode" : "5001000000000081",
  "Micw" : "FALSE"
}, {
  "PartitionNumber" : "3",
  "Name" : "New Partition_3",
  "SerialNumber" : "DE00000000_LL03",
  "NumSlots" : "85",
  "NumIOSlots" : "5",
  "NumDrives" : "1",
  "LunMasterDrive" : "3",
  "LunMasterDrivePhys" : "12",
  "LunMasterDriveArr" : [ "3" ],
  "LunMasterDrivePhysArr" : [ "12" ],
  "EncryptionMode" : "ISV",
  "BarcodeAlign" : "left",
  "BarcodeLength" : "8",
  "AutoClean" : "TRUE",
  "WWNode" : "5001000000000091",
  "Micw" : "FALSE"
} ]
```


2.2.6. Inventory of a partition

```
GET /rest/partition/inventory
```

Description

Deprecated: As of release 1.5.0.0-A00 this command should no longer be used. Please use /v1/logicalLibrary/inventory instead.

Returns the inventory of the selected partition including slots, drives and mailslots. It also returns the corresponding metadata which belongs to the different types of the elements

Parameters

Type	Name	Description	Schema
Query	**partitionNum** *required*	The partition number for which the inventory should be reported	integer

Responses

HTTP Code	Description	Schema
200	OK	Inventory
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficent rights	No Content
500	Error occurred at operation	Error

Produces

- `application/json`

Security

Type	Name
apiKey	**UserSecurity**

Example HTTP response

Response 200

```
{
  "Slots" : [ {
    "PhysicalNumber" : "1",
    "LogicalNumber" : "1.1",
    "Module" : "1",
    "Partition" : "1",
    "Mailslot" : "FALSE",
    "Cartridge" : "FALSE",
    "CartridgeType" : "N/A",
    "Access" : "TRUE",
    "Blocked" : "TRUE"
  }, {
    "PhysicalNumber" : "2",
    "LogicalNumber" : "1.2",
    "Module" : "1",
    "Partition" : "1",
```

```
    "Mailslot" : "FALSE",
    "Cartridge" : "FALSE",
    "CartridgeType" : "N/A",
    "Access" : "TRUE",
    "Blocked" : "FALSE"
  }, {
    "PhysicalNumber" : "3",
    "LogicalNumber" : "1.3",
    "Module" : "1",
    "Partition" : "1",
    "Mailslot" : "FALSE",
    "Cartridge" : "TRUE",
    "Barcode" : "TT0013L4",
    "CartridgeType" : "Data",
    "CartridgeSubType" : 0,
    "CartridgeGeneration" : "4",
    "CartridgeEncrypted" : "FALSE",
    "Access" : "TRUE",
    "Blocked" : "FALSE"
  }, { }, {
    "PhysicalNumber" : "280",
    "LogicalNumber" : "7.40",
    "Module" : "7",
    "Partition" : "0",
    "Mailslot" : "TRUE",
    "Cartridge" : "FALSE",
    "CartridgeType" : "N/A",
    "Access" : "TRUE",
    "Blocked" : "FALSE"
  } ],
  "Drives" : [ {
    "PhysicalNumber" : "10",
    "LogicalNumber" : "1",
    "Partition" : "0",
    "Vendor" : "VENDOR  ",
    "Product" : "Ultrium 7-SCSI  ",
    "FWRevision" : "ABCD",
    "SerialNumber" : 1234567890
  }, {
    "PhysicalNumber" : "11",
    "LogicalNumber" : "2",
    "Partition" : "0",
    "Vendor" : "VENDOR  ",
    "Product" : "Ultrium 7-SCSI  ",
    "FWRevision" : "ABCD",
    "SerialNumber" : 1234567891
  }, {
    "PhysicalNumber" : "12",
    "LogicalNumber" : "3",
    "Partition" : "0",
    "Vendor" : "VENDOR  ",
```

```
    "Product" : "Ultrium 7-SCSI  ",
    "FWRevision" : "ABCD",
    "SerialNumber" : 1234567892
  }, {
    "PhysicalNumber" : "20",
    "LogicalNumber" : "4",
    "Partition" : "0",
    "Vendor" : "",
    "Product" : "",
    "FWRevision" : "",
    "SerialNumber" : ""
  } ]
}
```

2.2.7. List of media of a partition

```
GET /rest/partition/mediainfo
```

Description

Deprecated: As of release 1.5.0.0-A00 this command should no longer be used. Please use /v1/logicalLibrary/mediainfo instead.

Returns the list of media which are assigned to the partition including detailed information about them

Parameters

Type	Name	Description	Schema
Query	**partitionNum** *required*	The partition number for which the media information should be reported	integer

Responses

HTTP Code	Description	Schema
200	OK	< MediaInfoData > array
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficent rights	No Content

HTTP Code	Description	Schema
500	Error occurred at operation	Error

Produces

- `application/json`

Security

Type	Name
apiKey	**UserSecurity**

Example HTTP response

Response 200

```
[ {
  "Barcode" : "TD239ML4",
  "LocationType" : "SLOT",
  "LogicalNumber" : "3.35",
  "PhysicalNumber" : "115",
  "Cleaning" : "FALSE",
  "Partition" : "2",
  "Generation" : "4",
  "SubType" : "0",
  "Protection" : "FALSE",
  "Encryption" : "FALSE",
  "NoLoads" : "0",
  "MBRead" : "0",
  "MBReadLoad" : "0",
  "MBWritten" : "0",
  "MBWrittenLoad" : "0"
}, {
  "Barcode" : "TC084ML5",
  "LocationType" : "SLOT",
  "LogicalNumber" : "5.10",
  "PhysicalNumber" : "170",
  "Cleaning" : "FALSE",
  "Partition" : "2",
  "Generation" : "5",
  "SubType" : "0",
  "Protection" : "TRUE",
  "Encryption" : "FALSE",
  "NoLoads" : "64",
  "MBRead" : "1643",
  "MBReadLoad" : "1603",
  "MBWritten" : "943",
  "MBWrittenLoad" : "943"
} ]
```


2.3. Library Information

Library Status / Information

2.3.1. Get physical libary information

```
GET /rest/library/baseinfo
```

Description

The "baseinfo" command returns information about the physical tape library like serial number, MAC address and other useful data

Responses

HTTP Code	Description	Schema
200	OK	LibraryInfo
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficent rights	No Content
500	Error occurred at operation	Error

Produces

- `application/json`

Security

Type	Name
apiKey	**UserSecurity**

Example HTTP response

Response 200


```
{
  "BaseInfo" : {
    "SerialNumber" : "DE00000000",
    "MacAdress_1" : "00:d0:93:3c:76:ee",
    "MacAdress_2" : "",
    "Vendor" : "VENDOR",
    "ProductID" : "ProductID",
    "BaseFWRevision" : "SSx1",
    "BaseFWBuildDate" : "2017-02-20",
    "ExpansionFWRevision" : "0.20",
    "WWNodeName" : "5001000000000000",
    "RoboticHWRevision" : "4",
    "RoboticFWRevision" : "0.10",
    "RoboticSerialNumber" : "564EA000103",
    "NoOfModules" : "7",
    "LibraryType" : "32"
  },
  "ModulesInfo" : [ {
    "PhysicalNumber" : "1",
    "LogicalNumber" : "1",
    "ReadyStatus" : "TRUE",
    "SerialNumber" : "DE56400022"
```

```
  }, {
    "PhysicalNumber" : "2",
    "LogicalNumber" : "2",
    "ReadyStatus" : "TRUE",
    "SerialNumber" : "DE56400041"
  }, {
    "PhysicalNumber" : "3",
    "LogicalNumber" : "3",
    "ReadyStatus" : "TRUE",
    "SerialNumber" : "DE56400040"
  }, {
    "PhysicalNumber" : "4",
    "LogicalNumber" : "4",
    "ReadyStatus" : "TRUE",
    "SerialNumber" : ""
  }, {
    "PhysicalNumber" : "5",
    "LogicalNumber" : "5",
    "ReadyStatus" : "TRUE",
    "SerialNumber" : "DE56400045"
  }, {
    "PhysicalNumber" : "6",
    "LogicalNumber" : "6",
    "ReadyStatus" : "TRUE",
    "SerialNumber" : "DE56400042"
  }, {
    "PhysicalNumber" : "7",
    "LogicalNumber" : "7",
    "ReadyStatus" : "TRUE",
    "SerialNumber" : "DE564000XX"
  } ]
}
```

2.3.2. Get list of recent events

```
GET /rest/library/getevents
```

Description

Based on the "EventType" this request returns a list of the recent events. The event types are
"Ticket"
"Info"
"Config"
"Service"
The number of returns can be limited by the parameter "MaxNum". To include already closed Tickets add "IncClosed" param.

Parameters

Type	Name	Description	Schema
Query	**EventType** *required*	The type of event	enum (Ticket, Info, Config, Service)
Query	**MaxNum** *optional*	The maximum number of returned events. If parameter is not given, all events available will be returned	integer
Query	**IncClosed** *optional*	The closed state of the ticket.	enum (TRUE, FALSE)

Responses

HTTP Code	Description	Schema
200	OK	< EventEntry > array
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficent rights	No Content
500	Error occurred at operation	Error

Produces

- `application/json`

Security

Type	Name
apiKey	**UserSecurity**

Example HTTP response

Response 200

```
[ {
  "Number" : "33",
  "TimeStamp" : "2017_03_23T15:22:34:636Z",
  "EventCode" : "9003",
  "Text" : "Move media command was executed",
  "State" : "TICKET_PENDING",
  "Closed" : "FALSE"
}, {
  "Number" : "32",
  "TimeStamp" : "2017_03_23T14:53:53:629Z",
  "EventCode" : "9028",
  "Text" : "Configuration backup to base module was successful",
  "State" : "TICKET_PENDING",
  "Closed" : "FALSE"
}, {
  "Number" : "31",
  "TimeStamp" : "2017_03_23T14:33:05:235Z",
  "EventCode" : "9026",
  "Text" : "Library test successfully finished",
  "State" : "TICKET_PENDING",
  "Closed" : "FALSE"
}, {
  "Number" : "30",
  "TimeStamp" : "2017_03_23T14:33:04:874Z",
  "EventCode" : "9003",
  "Text" : "Move media command was executed",
  "State" : "TICKET_PENDING",
  "Closed" : "FALSE"
}, {
  "Number" : "29",
  "TimeStamp" : "2017_03_23T14:32:18:319Z",
  "EventCode" : "9003",
  "Text" : "Move media command was executed",
  "State" : "TICKET_RESOLVED",
  "Closed" : "FALSE"
} ]
```


2.3.3. Mailslot Status

```
GET /rest/library/io/status
```

Description

Returns the status of the mailslots

Responses

HTTP Code	Description	Schema
200	OK	IOStatus
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficent rights	No Content
500	Error occurred at operation	Error

Produces

- `application/json`

Security

Type	Name
apiKey	**UserSecurity**

Example HTTP response

Response 200

```
[ {
  "ModuleNo" : "7",
  "Configured" : "FALSE",
  "OpenStatus" : "FALSE",
  "Unlocked" : "FALSE"
}, {
  "ModuleNo" : "6",
  "Configured" : "FALSE",
  "OpenStatus" : "FALSE",
  "Unlocked" : "FALSE"
}, {
  "ModuleNo" : "5",
  "Configured" : "TRUE",
  "OpenStatus" : "FALSE",
  "Unlocked" : "FALSE"
}, {
  "ModuleNo" : "4",
  "Configured" : "TRUE",
  "OpenStatus" : "FALSE",
  "Unlocked" : "FALSE"
}, {
  "ModuleNo" : "3",
  "Configured" : "FALSE",
  "OpenStatus" : "FALSE",
  "Unlocked" : "FALSE"
}, {
  "ModuleNo" : "2",
  "Configured" : "FALSE",
  "OpenStatus" : "FALSE",
  "Unlocked" : "FALSE"
}, {
  "ModuleNo" : "1",
  "Configured" : "FALSE",
  "OpenStatus" : "FALSE",
  "Unlocked" : "FALSE"
} ]
```


2.3.4. License List

```
GET /rest/library/license
```

Description

Returns a list of installed feature licenses

Responses

HTTP Code	Description	Schema
200	OK	< LicenseInfo > array
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficent rights	No Content

Security

Type	Name
apiKey	**AdminSecurity**

2.3.5. Get library logs dump

```
GET /rest/library/logs
```

Description

Generates and returns a log file from the stack in binary format "application/octet-stream". Other responses are returned as "application/json"

Responses

HTTP Code	Description	Schema
200	OK	No Content
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficent rights	No Content
500	Error occurred at operation	Error

Produces

- `application/json`
- `application/octet-stream`

Security

Type	Name
apiKey	**AdminSecurity**

2.3.6. Get physical library status

```
GET /rest/library/status
```

Description

The "status" command returns the live status of the physical library like health, number of moves and other live data

Responses

HTTP Code	Description	Schema
200	OK	LibraryStatus
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficent rights	No Content
500	Error occurred at operation	Error

Produces

- `application/json`

Security

Type	Name
apiKey	**UserSecurity**

Example HTTP response

Response 200

```
{
  "BaseStatus" : {
    "Information" : "Idle (Online)",
    "RobStatus" : "Idle",
    "MoveCount" : "1",
    "PowerUpCount" : "10",
    "PowerOnTime" : "16 days, 3 hours, 2 minutes",
    "LibHealth" : "WARNING"
  },
  "ModulesStatus" : [ {
    "PhysicalNumber" : "1",
    "LogicalNumber" : "1",
    "Health" : "OK"
  }, {
    "PhysicalNumber" : "2",
    "LogicalNumber" : "2",
    "Health" : "OK"
  }, {
    "PhysicalNumber" : "3",
    "LogicalNumber" : "3",
    "Health" : "OK"
  }, {
    "PhysicalNumber" : "4",
    "LogicalNumber" : "4",
    "Health" : "OK"
  }, {
    "PhysicalNumber" : "5",
    "LogicalNumber" : "5",
    "Health" : "OK"
  }, {
    "PhysicalNumber" : "6",
    "LogicalNumber" : "6",
    "Health" : "OK"
  }, {
    "PhysicalNumber" : "7",
    "LogicalNumber" : "7",
    "Health" : "OK"
  } ]
}
```


2.4. Library Operations

Library Operations

2.4.1. Inventory scan

```
POST /rest/library/inventory
```

Description

Executes an Inventory scan on the library

Responses

HTTP Code	Description	Schema
200	OK	No Content
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficent rights	No Content
500	Error occurred at operation	Error

Security

Type	Name
apiKey	**AdminSecurity**

2.4.2. Open Mailslot

```
POST /rest/library/io/open
```

Description

This command sends a request to open a mailslot. The mailslot block which is addressed has to be configured as mailslot first. The mailslot stays open for a defined time and will be locked automatically after this time. The status of the mailslot can be retrieved by issuing the IO/status request

Parameters

Type	Name	Description	Schema
Body	**Module** *required*	the logical module number	Module

Module

Name	Description	Schema
module *required*	**Minimum value** : 1 **Maximum value** : 7 **Example** : 1	integer

Responses

HTTP Code	Description	Schema
200	The mailslot open was successful	No Content
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficent rights	No Content
500	Error occurred at operation	Error

Consumes

- `application/json`

Security

Type	Name
apiKey	**AdminSecurity**

2.4.3. Move a cartridge

```
POST /rest/library/movemedia
```

Description

A cartridge can be moved from a slot or drive to another slot or drive.

Parameters

Type	Name	Description	Schema
Body	**Elements** *required*	the source element from where a cartridge should be taken. It has the form <Element type> and <Address>. The type can be "Slot" or "Drive". Depending on the type the Address will have the form "Logical Slot Number (m.s)" or "Logical Drivenumber"	Elements

Elements

Name	Description	Schema
SrcType *required*	**Example** : `"Slot"`	string
SrcAddress *required*	**Example** : `"1.2"`	string
DestType *required*	**Example** : `"Drive"`	string
DestAddress *required*	**Example** : `"3"`	string

Responses

HTTP Code	Description	Schema
200	The move was executed successfully	No Content
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficent rights	No Content
500	Error occurred at operation	Error

Consumes

- `application/json`

Security

Type	Name
apiKey	**AdminSecurity**

2.4.4. Shut down the library

```
POST /rest/library/powerdown
```

Description

The library will be shut down. During shutdown the robotic will be secured at a safe transport position. The shutdown will be initiated after response to this command

Parameters

Type	Name	Schema
Body	**RoboticPos** *required*	RoboticPos

RoboticPos

Name	Description	Schema
RoboticPos *optional*	The position of the robotic after the shutdown."SHIPPING_POS" moves the robotic to the standard shipping position in the base module and activates the shipping lock. "RACK_SHIPPING_POS" moves the robotic to the bottom of the library independent of the base module position (shipment in rack position). "PARK_POS" moves the robotic to the default park position. **Example** : `"SHIPPING_POS"`	enum (SHIPPING_POS, RACK_SHIPPING_POS, PARK_POS)

Responses

HTTP Code	Description	Schema
200	The shutdown was executed successfully	No Content
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficent rights	No Content
500	Error occurred at operation	Error

Security

Type	Name
apiKey	**AdminSecurity**

2.4.5. Reboot the libary

```
POST /rest/library/reboot
```

Description

The library will be rebooted. The shutdown will be initiated after response to this command. The connection will be lost until the library restarted successfully. A new login will be required

Responses

HTTP Code	Description	Schema
200	The reboot was executed successfully	No Content
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficent rights	No Content
500	Error occurred at operation	Error

Security

Type	Name
apiKey	**AdminSecurity**

2.5. Library Configuration

Library Configuration

2.5.1. Mailslot Operations

```
POST /rest/library/io/config
```

Description

With this command the mailslot block in a magazine can be configured as mailslots or as standard storage slots. In order to be able to open mailslot slots of the magazine block they need to be configured as mailslots

Parameters

Type	Name	Description	Schema
Body	**mailslotConf** *required*	the logical module number and mode of mailslot. A TRUE will configure mailslot mode, a FALSE storage mode	mailslotConf

mailslotConf

Name	Description	Schema
module *required*	**Minimum value** : 1 **Maximum value** : 7	integer
mode *required*		enum (TRUE, FALSE)

Responses

HTTP Code	Description	Schema
200	The mailslot configuration was successful	No Content
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficent rights	No Content
500	Error occurred at operation	Error

Security

Type	Name
apiKey	**AdminSecurity**

2.5.2. Import License key

```
POST /rest/library/license
```

Description

With this command keys for licensed features can be imported and added to the system. The key inherits the feature and additional parameters which have to licensed

Parameters

Type	Name	Description	Schema
Body	**LicenseKey** *required*	the license key to add	LicenseKey

LicenseKey

Name	Description	Schema
licenseKey *required*	**Minimum length** : 15	string

Responses

HTTP Code	Description	Schema
200	The license key was added succesfully	No Content
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficent rights	No Content
500	Error occurred at operation	Error

Security

Type	Name
apiKey	**AdminSecurity**

2.5.3. Change MAC of the Library

```
POST /rest/library/mac
```

Description

With this command the MACs can be set. After successful change the Library will be rebooted.

Parameters

Type	Name	Description	Schema
Body	**setConf** *required*	the new MACs to be set	setConf

setConf

Name	Description	Schema
mac0 *optional*	**Length** : 17	string
mac1 *optional*	**Length** : 17	string

Responses

HTTP Code	Description	Schema
200	The settings succesfully changed.	No Content
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficent rights	No Content
500	Error occurred at operation	Error

Security

Type	Name
apiKey	**ServiceSecurity**

2.5.4. Change MAC and/or WWN of the Library

```
POST /rest/library/mac_wwn
```

Description

With this command the WWN (WWide Node Name) and the MACs can be configured. After successful change the Library will be rebooted.

Parameters

Type	Name	Description	Schema
Body	**setConf** *required*	the new WWN and/or MACs to be set	setConf

setConf

Name	Description	Schema
wwn *optional*	**Length** : 16	string
mac0 *optional*	**Length** : 17	string
mac1 *optional*	**Length** : 17	string

Responses

HTTP Code	Description	Schema
200	The settings succesfully changed.	No Content
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficent rights	No Content
500	Error occurred at operation	Error

Security

Type	Name
apiKey	**ServiceSecurity**

2.5.5. Reset the Configuration

```
POST /rest/library/resetconfig
```

Description

After issuing this command the configuration of the library will be set to standard defaults or to manufacturing defaults depending on the ManufacturingMode parameter. The standard defaults will keep IP address, counters and manufacturing defaults will reset everything. In case of manual configured network the connection will possibly be lost. If no Mode parameter is submitted standard default mode will be used. Reset to Manufacturing Mode will also delete existing license keys in the system

Parameters

Type	Name	Schema
Body	**ManufacturingMode** *optional*	ManufacturingMode

ManufacturingMode

Name	Description	Schema
ManufacturingMode *optional*	The mode of reset. "TRUE" will do a full reset including IP settings and licenses **Example** : `"FALSE"`	enum (TRUE, FALSE)
RoboticPos *optional*	The position of the robotic after the shutdown."SHIPPING_POS" moves the robotic to the standard shipping position in the base module and activates the shipping lock. "RACK_SHIPPING_POS" moves the robotic to the bottom of the library independent of the base module position (shipment in rack position). "PARK_POS" moves the robotic to the default park position. **Example** : `"SHIPPING_POS"`	enum (SHIPPING_POS, RACK_SHIPPING_POS, PARK_POS)
PowerDown *optional*	"TRUE" powers the unit off after a reset. "FALSE" keeps default reboot behavior	enum (TRUE, FALSE)

Responses

HTTP Code	Description	Schema
200	The reset configuration was executed successfully	No Content
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficent rights	No Content

HTTP Code	Description	Schema
500	Error occurred at operation	Error

Consumes

- `application/json`

Security

Type	Name
apiKey	**AdminSecurity**
apiKey	**ServiceSecurity**

2.5.6. Set library serial number

```
PUT /rest/library/serialnum
```

Description

The library serial number can be set.

Note: After setting the serial number the library must be restarted for the changes to take effect.

Parameters

Type	Name	Schema
Body	**SerialNumber** *required*	SerialNumber

SerialNumber

Name	Description	Schema
serialnum *required*	**Length** : `10 - 14` **Example** : `"DE1234567890"`	string

Responses

HTTP Code	Description	Schema
200	OK	No Content
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficent rights	No Content
500	Error occurred at operation	Error

Consumes

- `application/json`

Security

Type	Name
apiKey	**ServiceSecurity**

2.5.7. Upgrade Library SW

```
POST /rest/library/swupgrade
```

Description

The command upgrades the SW on the library. It returns before the SW upgrade process is started. After the upgrade a reboot will be initiated. The connection will be lost during this time frame

Parameters

Type	Name	Description	Schema
FormData	**SWImage** *required*	The SW upgrade file	file

Responses

HTTP Code	Description	Schema
200	OK	No Content
401	Access denied - not authenticated	No Content

HTTP Code	Description	Schema
403	Access forbidden - insufficent rights	No Content
500	Error occurred at operation	Error

Consumes

- `multipart/form-data`

Security

Type	Name
apiKey	**AdminSecurity**

2.5.8. Change WWN of the Library

```
POST /rest/library/wwn
```

Description

With this command the WWN (WWide Node Name) can be set. Format is 16 hex characters [0-9 A-F]. After successful change the Library will be rebooted.

Parameters

Type	Name	Description	Schema
Body	**setConf** *required*	the new WWN to be set	setConf

setConf

Name	Description	Schema
wwn *required*	**Length** : 16	string

Responses

HTTP Code	Description	Schema
200	The WWN was succesfully changed.	No Content

HTTP Code	Description	Schema
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficent rights	No Content
500	Error occurred at operation	Error

Security

Type	Name
apiKey	**ServiceSecurity**

2.6. Library Test

Library Tests

2.6.1. Slot to Slot Test

```
POST /rest/test/slot2slot
```

Description

Performs the slot to slot test n times. The number of executions has to be defined in the count parameter. The command returns immediately after starting the test. The progress has to be polled by the /test/status request

Parameters

Type	Name	Description	Schema
Body	**count** *required*	the number of test iterations to perform	count

count

Name	Description	Schema
count *required*	**Maximum value** : `100` **Example** : `10`	integer

Responses

HTTP Code	Description	Schema
200	The test could be started succesfully	No Content
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficent rights	No Content
500	Error occurred at operation	Error

Consumes

- `application/json`

Security

Type	Name
apiKey	**AdminSecurity**

2.6.2. Test Status

```
GET /rest/test/status
```

Description

Returns the status of a test which is currently executed. A test run always asynchronously, a response is sent immediately after the start. This function is used to monitor the status of the test, it returns the current test cycle and the run status.

Responses

HTTP Code	Description	Schema
200	Status of current test	TestStatus
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficent rights	No Content
500	Error occurred at operation	Error

Produces

- `application/json`

Security

Type	Name
apiKey	**AdminSecurity**

Example HTTP response

Response 200

```
{
  "currentCycle" : "2",
  "testCycles" : "10",
  "status" : "TRUE",
  "running" : "FALSE"
}
```

2.6.3. System (Demo) Test

```
POST /rest/test/systemtest
```

Description

Performs the integrated system (Demo) test n times. The number of executions has to be defined in the count parameter. The command returns immediately after starting the test. The progress has to be polled by the /test/status request

Parameters

Type	Name	Description	Schema
Body	**testParameter** *required*	the number of test iterations to perform and seating mode	testParameter

testParameter

Name	Description	Schema
count *required*	**Maximum value** : `100` **Example** : `10`	integer

Name	Description	Schema
seating *optional*		enum (TRUE, FALSE)

Responses

HTTP Code	Description	Schema
200	The test could be started succesfully	No Content
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficent rights	No Content
500	Error occurred at operation	Error

Security

Type	Name
apiKey	**AdminSecurity**

2.7. Drive Configuration

Drive Configuration

2.7.1. Upgrade Drive FW

```
POST /rest/drive/fwupgrade
```

Description

The command upgrades the FW of the specified drives in the library.

Parameters

Type	Name	Description	Schema
FormData	**FWImage** *required*	The FW upgrade file	file
FormData	**drivenumber** *required*	The logical drive number(s), multiple drives possible as array (comma separated values)	< integer > array(csv)

Responses

HTTP Code	Description	Schema
200	The FW upgrade was successful	No Content
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficent rights	No Content
500	Error occurred at operation	Error

Consumes

- `multipart/form-data`

Security

Type	Name
apiKey	**AdminSecurity**

2.7.2. Drive Reset

```
POST /rest/drive/reset
```

Description

The command performs a drive reset

Parameters

Type	Name	Description	Schema
Body	**DriveNumber** *required*	The logical number of the drive to be reset	DriveNumber

DriveNumber

Name	Description	Schema
DriveNumber *required*	**Minimum value** : 1 **Maximum value** : 21 **Example** : 3	integer

Responses

HTTP Code	Description	Schema
200	The drive was reset successfully	No Content
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficent rights	No Content
500	Error occurred at operation	Error

Consumes

- `application/json`

Security

Type	Name
apiKey	**AdminSecurity**

2.8. V1-Authentication

Authentication request

2.8.1. Login Request

```
POST /v1/login
```

Description

Required Initial request to get authenticated against the tape library. The user roles/names of the library are used. This API command returns after a successful authentication an session based Bearer token, which has to be attached in the "Authorization" header of every following API command during a session.

Parameters

Type	Name	Description	Schema
Body	**user** *required*	the user/password which should be used to log in.	user

user

Name	Description	Schema
username *required*	**Example** : `"administrator"`	string
password *required*	**Example** : `"password"`	string
service_password *optional*	service password in case the administrator password is additionally required **Example** : `"servicepwd"`	string
change_password *optional*	If account needs a password change after valid login credentials provided this param needs to be set with the new password	string

Responses

HTTP Code	Description	Schema
201	Login request was successful and session token is generated	Response 201
401	Access denied - not authenticated	No Content
500	Error occurred at operation	Error

Response 201

Name	Schema
token *required*	string

Consumes

- `application/json`

Example HTTP response

Response 201

```
{
  "status" : "ok",
  "token" : "Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpYXQiOjEwMDg5OSwiZXhwIjoxMDgwOTksIlVST0xFIjoyL
CJqaXQiOiI0NmEzOTBiZDcwMTA5Y2ViYjk0MzExNTQ2YjkxYjI3ZXQyM2Y1Y2hhYmE5bnNpdTQ3bnU3cXI1a3Q
zMDAwMThhMjM2ZDQ5MDYuNTg2OTc4MTcifQ.DrCpCaX0jMwREFRYmSHazGY7wOs6QfGzRcVud3cDjA0"
}
```


2.9. V1-Accessor

Accessor requests

2.9.1. Get Accessors

```
GET /v1/accessors
```

Description

Retrieve a list of accessors or a single accessor resource.

Responses

HTTP Code	Description	Schema
200	OK	V1Accessor
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficent rights	No Content
500	Error occurred at operation	Error

Produces

- `application/json`

Security

Type	Name
apiKey	**AdminSecurity**

Example HTTP response

Response 200

```
{
  "location" : "accessor_Aa",
  "state" : "online",
  "moves" : 1000
}
```

2.10. V1-Drives

Drives requests

2.10.1. Get Drives

```
GET /v1/drive
```

Description

Retrieve a list of drives or a single drive resource.

Responses

HTTP Code	Description	Schema
200	OK	V1DriveInfo
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficent rights	No Content
500	Error occurred at operation	Error

Produces

- `application/json`

Security

Type	Name
apiKey	**AdminSecurity**

Example HTTP response

Response 200


```
[ {
  "location" : "drive_M1R2",
  "sn" : "0000000D31",
  "mfgSn" : "1234567890",
  "mediaType" : "LTO",
  "state" : "unloaded",
  "mtm" : "ULT3580-HH8",
  "interface" : "fibreChannel",
  "logicalLibrary" : "LogicalLib_1",
  "use" : "controlPath",
  "firmware" : "M2DF",
  "encryption" : "disabled",
  "mounts" : 513,
  "barcode" : "1234567890",
  "wwnn" : "5001000000000D31",
  "elementAddress" : 1
}, {
  "location" : "drive_M2R1",
  "sn" : "0000000D45",
  "mfgSn" : "1234567891",
  "mediaType" : "LTO",
  "state" : "unloaded",
  "mtm" : "ULT3580-HH8",
  "interface" : "fibreChannel",
  "logicalLibrary" : "LogicalLib_1",
  "use" : "access",
  "firmware" : "M2DF",
  "encryption" : "disabled",
  "mounts" : 295,
  "barcode" : "1234567891",
  "wwnn" : "5001000000000D45",
  "elementAddress" : 2
} ]
```


2.10.2. Get drive information

```
GET /v1/drive/information
```

Description

Returns the information about drives like Product Id, Model, Serialno and other useful information.

Responses

HTTP Code	Description	Schema
200	OK	< V1DriveInfoNaming > array

Security

Type	Name
apiKey	**UserSecurity**

Example HTTP response

Response 200


```
{
  "LogicalNumber" : "1",
  "PhysicalNumber" : "10",
  "Module" : "1",
  "LogicalLibrary" : "0",
  "Generation" : "8",
  "Cartridge" : "FALSE",
  "Vendor" : "IBM",
  "Product" : "ULT3580-HH8",
  "FWRevision" : "J28F",
  "SerialNumber" : "000000005B",
  "WWNodeName" : "",
  "Interface" : "FC",
  "MFGSerialNumber" : "1013000188",
  "ErrorState" : "TRUE",
  "Power" : "FALSE",
  "Presence" : "TRUE",
  "ADTMode" : "IADT"
}
```


2.10.3. Get Drives by <location> or <sn>

```
GET /v1/drive/{location OR sn}
```

Description

Retrieve a list of drives or a single drive resource.

Parameters

Type	Name	Description	Schema
Path	**location OR sn** *required*	**location** A string representing the unique location of the tape drive. This is also the unique identifier for the drive. The format is "drive_M<m>R<r>" where 'm' is the module and 'r' is the row or module location (1 through 3 within the module) the drive is installed in. **sn** A string representing the serial number of the drive. This can also be used as a unique identifier for the tape drive.	string

Responses

HTTP Code	Description	Schema
200	OK	V1DriveInfo
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficent rights	No Content
500	Error occurred at operation	Error

Produces

- `application/json`

Security

Type	Name
apiKey	**AdminSecurity**

2.10.4. Reset Drive

```
POST /v1/drive/{location}/reset
```

Description

Execute a drive reset on the given drive.

Parameters

Type	Name	Description	Schema
Path	**location** *required*	location A string representing the unique location of the tape drive. This is also the unique identifier for the drive. The format is "drive_M<m>R<r>" where 'm' is the module and 'r' is the row or module location (1 through 3 within the module) the drive is installed in.	string

Responses

HTTP Code	Description	Schema
200	OK	No Content
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficent rights	No Content
500	Error occurred at operation	Error

Produces

- `application/json`

Security

Type	Name
apiKey	**AdminSecurity**

2.11. V1-Ethernet

Ethernet Ports requests

2.11.1. Get Ethernet Ports

```
GET /v1/ethernetPorts
```

Description

Retrieve a list of Ethernet ports or a single Ethernet port resource.

Responses

HTTP Code	Description	Schema
200	OK	V1EthernetPort
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficent rights	No Content
500	Error occurred at operation	Error

Produces

- `application/json`

Security

Type	Name
apiKey	**AdminSecurity**

Example HTTP response

Response 200

```
[ {
  "location" : "ethernetPort_M1P1",
  "macAddress" : "AB:CD:93:21:5F:EE",
  "ipv4Address" : "172.16.25.93",
  "ipv4Subnet" : "255.255.248.0",
  "ipv4Gateway" : "172.16.31.254",
  "ipv4Assignment" : "dynamic",
  "ipv4Primary" : "172.16.31.254"
} ]
```

2.11.2. Get Ethernet Port by location

```
GET /v1/ethernetPorts/{location}
```

Description

Retrieve a single Ethernet port resource.

Parameters

Type	Name	Description	Schema
Path	**location** *required*	A string representing the unique location of this port. The format of this field is "ethernetPort_M<m>P<p>" where <m> is the module and <p> is the port number.	string

Responses

HTTP Code	Description	Schema
200	OK	V1EthernetPort
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficent rights	No Content
500	Error occurred at operation	Error

Produces

- `application/json`

Security

Type	Name
apiKey	**AdminSecurity**

2.12. V1-Events

Events requests

2.12.1. Get Events

```
GET /v1/events
```

Description

The library events that track all resource state changes within the library.

Responses

HTTP Code	Description	Schema
200	OK	V1EventEntry
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficent rights	No Content
500	Error occurred at operation	Error

Produces

- `application/json`

Security

Type	Name
apiKey	**AdminSecurity**

Example HTTP response

Response 200

```
[ {
  "ID" : "I453",
  "severity" : "information",
  "time" : "2020-03-18T14:28:45+0100",
  "type" : "information",
  "location" : "library",
  "description" : "Library test stopped by user
(CMD_NAME=SYSTEM_TEST,CMD_INITIATOR=OCP_GUI_CTRL,PARAM1=10,LOAD_UNLOAD_HOLD=TRUE,PHY_N
UM=1,SYS_COMPONENT=SYSTEM,CYCLE_COUNT=1)",
  "errorCode" : "9027"
}, {
  "ID" : "I452",
  "severity" : "information",
  "time" : "2020-03-18T14:28:44+0100",
  "type" : "information",
  "location" : "accessor_Aa",
  "description" : "Move cartridge command was executed
(SRC_ELEM_ADDR=11,SRC_ELEM_TYPE=4,DST_ELEM_ADDR=199,DST_ELEM_TYPE=2,MCHG_MEDI_PVT=CH01
82L8,SYS_COMPONENT=ROBOTICS,PHY_NUM=1)",
  "errorCode" : "9003"
} ]
```


2.12.2. Get Events by ID

```
GET /v1/events/{ID}
```

Description

The library events that track all resource state changes within the library.

Parameters

Type	Name	Description	Schema
Path	**ID** *required*	A unique identifier of the event in the library.	string

Responses

HTTP Code	Description	Schema
200	OK	V1EventEntry
401	Access denied - not authenticated	No Content

HTTP Code	Description	Schema
403	Access forbidden - insufficent rights	No Content
500	Error occurred at operation	Error

Produces

- `application/json`

Security

Type	Name
apiKey	**AdminSecurity**

2.13. V1-Library

Library requests

2.13.1. Get Library

```
GET /v1/library
```

Description

The general information about the tape library and its settings. This is a singleton object.

Responses

HTTP Code	Description	Schema
200	OK	V1LibraryEntry
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficent rights	No Content
500	Error occurred at operation	Error

Produces

- `application/json`

Security

Type	Name
apiKey	**AdminSecurity**

Example HTTP response

Response 200

```
{
  "status" : "WARNING",
  "totalCapacity" : 80,
  "licensedCapacity" : 80,
  "totalCartridges" : 31,
  "assignedCartridges" : 10,
  "firmware" : "1.3.1.0-S13",
  "sn" : "DE13040723",
  "time" : "2020-03-18T14:44:01+0100",
  "secureCommunications" : "disabled"
}
```

2.13.2. Calibrate Library

```
POST /v1/library/calibrate
```

Description

Execute a calibration the library. This does not return until the calibration is finished.

Responses

HTTP Code	Description	Schema
200	OK	No Content
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficent rights	No Content
500	Error occurred at operation	Error

Produces

- `application/json`

Security

Type	Name
apiKey	**AdminSecurity**

2.13.3. Inventory Library

```
POST /v1/library/inventory
```

Description

Execute an inventory scan on the library. This does not return until the inventory is finished.

Responses

HTTP Code	Description	Schema
200	OK	No Content
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficent rights	No Content
500	Error occurred at operation	Error

Produces

- `application/json`

Security

Type	Name
apiKey	**AdminSecurity**

2.13.4. Get Inventory of physical library

```
GET /v1/library/inventory
```

Description

Returns the inventory of the physical library including slots, drives and I/O stations. It also returns the corresponding metadata which belongs to the different types of the elements

Responses

HTTP Code	Description	Schema
200	OK	V1Inventory
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficent rights	No Content
500	Error occurred at operation	Error

Produces

- `application/json`

Security

Type	Name
apiKey	**UserSecurity**

Example HTTP response

Response 200


```
{
  "Slots" : [ {
    "PhysicalNumber" : 1,
    "LogicalNumber" : "1.1",
    "Module" : 1,
    "LogicalLibrary" : 1,
    "Mailslot" : true,
    "Cartridge" : false,
    "CartridgeType" : "N/A",
    "CartridgeSubType" : 0,
    "Access" : true,
    "Blocked" : false
  }, {
    "PhysicalNumber" : 2,
    "LogicalNumber" : "1.2",
    "Module" : 1,
    "LogicalLibrary" : 1,
    "Mailslot" : false,
    "Cartridge" : false,
    "CartridgeType" : "N/A",
    "CartridgeSubType" : 0,
    "Access" : true,
```

```
    "Blocked" : false
  }, {
    "PhysicalNumber" : 3,
    "LogicalNumber" : "1.3",
    "Module" : 1,
    "LogicalLibrary" : 1,
    "Mailslot" : false,
    "Cartridge" : true,
    "Barcode" : "TT0013L4",
    "CartridgeType" : "Data",
    "CartridgeSubType" : 0,
    "CartridgeGeneration" : 4,
    "CartridgeEncrypted" : false,
    "Access" : true,
    "Blocked" : false
  }, { }, {
    "PhysicalNumber" : 280,
    "LogicalNumber" : "7.40",
    "Module" : 7,
    "LogicalLibrary" : 0,
    "Mailslot" : true,
    "Cartridge" : false,
    "CartridgeType" : "N/A",
    "CartridgeSubType" : 0,
    "Access" : true,
    "Blocked" : false
  } ],
  "Drives" : [ {
    "PhysicalNumber" : 10,
    "LogicalNumber" : 1,
    "LogicalLibrary" : 0,
    "Vendor" : "VENDOR",
    "Product" : "Ultrium 7-SCSI",
    "FWRevision" : "ABCD",
    "SerialNumber" : 1234567890
  }, {
    "PhysicalNumber" : 11,
    "LogicalNumber" : 2,
    "LogicalLibrary" : 0,
    "Vendor" : "VENDOR",
    "Product" : "Ultrium 7-SCSI  ",
    "FWRevision" : "ABCD",
    "SerialNumber" : 1234567891
  }, {
    "PhysicalNumber" : 12,
    "LogicalNumber" : 3,
    "LogicalLibrary" : 0,
    "Vendor" : "VENDOR",
    "Product" : "Ultrium 7-SCSI",
    "FWRevision" : "ABCD",
    "SerialNumber" : 1234567892
```

```
  }, {
    "PhysicalNumber" : 20,
    "LogicalNumber" : 4,
    "LogicalLibrary" : 0,
    "Vendor" : "",
    "Product" : "",
    "FWRevision" : "",
    "SerialNumber" : ""
  } ]
}
```

2.13.5. List of media

```
GET /v1/library/mediainfo
```

Description

Returns the list of media which are located in the library including detailed information about them

Responses

HTTP Code	Description	Schema
200	OK	< V1MediaInfoData > array
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficent rights	No Content
500	Error occurred at operation	Error

Produces

- `application/json`

Security

Type	Name
apiKey	**UserSecurity**

Example HTTP response

Response 200

```
[ {
  "Barcode" : "TC070ML5",
  "LocationType" : "SLOT",
  "LogicalNumber" : "2.34",
  "PhysicalNumber" : 74,
  "Cleaning" : false,
  "LogicalLibrary" : 1,
  "Generation" : 5,
  "SubType" : 0,
  "Protection" : false,
  "Encryption" : false,
  "NoLoads" : 148,
  "MBRead" : 153,
  "MBReadLoad" : 150,
  "MBWritten" : 226,
  "MBWrittenLoad" : 226
}, {
  "Barcode" : "000113L5",
  "LocationType" : "SLOT",
  "LogicalNumber" : "3.4",
  "PhysicalNumber" : 84,
  "Cleaning" : false,
  "LogicalLibrary" : 1,
  "Generation" : 5,
  "SubType" : 0,
  "Protection" : false,
  "Encryption" : false,
  "NoLoads" : 11,
  "MBRead" : 0,
  "MBReadLoad" : 0,
  "MBWritten" : 14,
  "MBWrittenLoad" : 14
}, {
  "Barcode" : "TC084ML5",
  "LocationType" : "SLOT",
  "LogicalNumber" : "3.28",
  "PhysicalNumber" : 108,
  "Cleaning" : false,
  "Partition" : 2,
  "Generation" : 5,
  "SubType" : 0,
  "Protection" : true,
  "Encryption" : false,
  "NoLoads" : 64,
  "MBRead" : 1643,
  "MBReadLoad" : 1603,
  "MBWritten" : 943,
  "MBWrittenLoad" : 943
} ]
```

2.13.6. Reset Library

```
POST /v1/library/reset
```

Description

Execute a library reset. While a REST API response will always be created, the SCSI Read Buffer method of querying for that response will not necessarily occur before the library reset has begun. However, the buffer will be non-volatile and query-able after the reset is complete.

Responses

HTTP Code	Description	Schema
200	OK	No Content
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficent rights	No Content
500	Error occurred at operation	Error

Produces

- `application/json`

Security

Type	Name
apiKey	**AdminSecurity**

2.14. V1-LogicalLibrary

Logical Library Information

2.14.1. Create logical libraries in simple mode

```
POST /v1/logicalLibrary/createsimple
```

Description

Creates the number of logical libraries which are requested in the parameters. This commands corresponds to the Basic Wizard in RMI to create logical libraries. The maximum number of logical

libraries which can be created are limited by the number of installed drives.

Parameters

Type	Name	Description	Schema
Body	**logicalLibraryParameters** *required*	the parameters to create the logical libraries	logicalLibraryParameters

logicalLibraryParameters

Name	Description	Schema
numLogicalLibraries *required*	The number of logical libraries to create **Minimum value** : 1 **Maximum value** : 21 **Example** : 1	integer
barcodeLength *optional*	**Minimum value** : 6 **Maximum value** : 16 **Example** : 8	integer
barcodeAlignment *optional*	Alignment of the barcode for SCSI element status. Can be "left" or "right" **Default** : "left"	enum (left, right)
autoClean *optional*	Auto Cleaning for this logical library. 'TRUE' for enabling 'FALSE' to disable it. **Default** : "FALSE"	enum (TRUE, FALSE)
sequentialMode *optional*	Enable Sequential Mode for this logical library. 'TRUE' for enabling 'FALSE' to disable it. **Default** : "FALSE"	enum (TRUE, FALSE)
sequentialModeLoop *optional*	Restarting sequence when finished. Requires sequentialMode to be 'TRUE' **Default** : "FALSE"	enum (TRUE, FALSE)
sequentialModeAutoload *optional*	Load first cartridge automatically in sequential mode. Requires sequentialMode to be 'TRUE' **Default** : "FALSE"	enum (TRUE, FALSE)

Responses

HTTP Code	Description	Schema
200	OK	No Content
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficent rights	No Content
500	Error occurred at operation	Error

Consumes

- `application/json`

Security

Type	Name
apiKey	**AdminSecurity**

2.14.2. Get Logical Library information

```
GET /v1/logicalLibrary/information
```

Description

The command returns the list of logical libraries including logical library details

Responses

HTTP Code	Description	Schema
200	OK	< V1LogicalLibraryInfo > array
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficent rights	No Content
500	Error occurred at operation	Error

Produces

- `application/json`

Security

Type	Name
apiKey	**UserSecurity**

Example HTTP response

Response 200


```
[ {
  "LogicalLibraryNumber" : 1,
  "Name" : "logicalLibrary_1",
  "SerialNumber" : "DE00000000_LL01",
  "NumSlots" : 87,
  "NumIOSlots" : 0,
  "NumDrives" : 1,
  "LunPrimaryDrive" : 1,
  "LunPrimaryDrivePhys" : 10,
  "LunPrimaryDriveArr" : [ "1" ],
  "LunPrimaryDrivePhysArr" : [ "10" ],
  "EncryptionMode" : "ISV",
  "BarcodeAlign" : "left",
  "BarcodeLength" : 8,
  "AutoClean" : true,
  "WWNode" : "5001000000000071",
  "Micw" : false
}, {
  "LogicalLibraryNumber" : 2,
  "Name" : "logicalLibrary_1",
  "SerialNumber" : "DE00000000_LL02",
  "NumSlots" : 90,
  "NumIOSlots" : 5,
  "NumDrives" : 1,
  "LunPrimaryDrive" : 2,
  "LunPrimaryDrivePhys" : 11,
  "LunPrimaryDriveArr" : [ "2" ],
  "LunPrimaryDrivePhysArr" : [ "11" ],
  "EncryptionMode" : "ISV",
  "BarcodeAlign" : "left",
  "BarcodeLength" : 8,
  "AutoClean" : true,
  "WWNode" : "5001000000000081",
  "Micw" : false
} ]
```

2.14.3. Inventory of a logical library

```
GET /v1/logicalLibrary/inventory
```

Description

Returns the inventory of the selected logical library including slots, drives and mailslots. It also returns the corresponding metadata which belongs to the different types of the elements

Parameters

Type	Name	Description	Schema
Query	**logicalLibraryNum** *required*	The logical library number for which the inventory should be reported	integer

Responses

HTTP Code	Description	Schema
200	OK	V1Inventory
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficent rights	No Content
500	Error occurred at operation	Error

Produces

- `application/json`

Security

Type	Name
apiKey	**UserSecurity**

Example HTTP response

Response 200

```
{
  "Slots" : [ {
```

```
    "PhysicalNumber" : 1,
    "LogicalNumber" : "1.1",
    "Module" : 1,
    "LogicalLibrary" : 1,
    "Mailslot" : false,
    "Cartridge" : false,
    "CartridgeType" : "N/A",
    "Access" : true,
    "Blocked" : true
  }, {
    "PhysicalNumber" : 2,
    "LogicalNumber" : "1.2",
    "Module" : 1,
    "LogicalLibrary" : 1,
    "Mailslot" : false,
    "Cartridge" : false,
    "CartridgeType" : "N/A",
    "Access" : true,
    "Blocked" : false
  }, {
    "PhysicalNumber" : 3,
    "LogicalNumber" : "1.3",
    "Module" : 1,
    "LogicalLibrary" : 1,
    "Mailslot" : false,
    "Cartridge" : true,
    "Barcode" : "TT0013L4",
    "CartridgeType" : "Data",
    "CartridgeSubType" : 0,
    "CartridgeGeneration" : 4,
    "CartridgeEncrypted" : false,
    "Access" : true,
    "Blocked" : false
  }, { }, {
    "PhysicalNumber" : 280,
    "LogicalNumber" : "7.40",
    "Module" : 7,
    "LogicalLibrary" : 0,
    "Mailslot" : true,
    "Cartridge" : false,
    "CartridgeType" : "N/A",
    "Access" : true,
    "Blocked" : false
  } ],
  "Drives" : [ {
    "PhysicalNumber" : 10,
    "LogicalNumber" : 1,
    "LogicalLibrary" : 0,
    "Vendor" : "VENDOR",
    "Product" : "Ultrium 7-SCSI",
    "FWRevision" : "ABCD",
```

```
    "SerialNumber" : 1234567890
  }, {
    "PhysicalNumber" : 11,
    "LogicalNumber" : 2,
    "LogicalLibrary" : 0,
    "Vendor" : "VENDOR",
    "Product" : "Ultrium 7-SCSI",
    "FWRevision" : "ABCD",
    "SerialNumber" : 1234567891
  }, {
    "PhysicalNumber" : 12,
    "LogicalNumber" : 3,
    "LogicalLibrary" : 0,
    "Vendor" : "VENDOR",
    "Product" : "Ultrium 7-SCSI",
    "FWRevision" : "ABCD",
    "SerialNumber" : 1234567892
  }, {
    "PhysicalNumber" : 20,
    "LogicalNumber" : 4,
    "LogicalLibrary" : 0,
    "Vendor" : "",
    "Product" : "",
    "FWRevision" : "",
    "SerialNumber" : ""
  } ]
}
```


2.14.4. List of media of a logical library

```
GET /v1/logicalLibrary/mediainfo
```

Description

Returns the list of media which are assigned to the logical library including detailed information about them

Parameters

Type	Name	Description	Schema
Query	**logicalLibraryNum** *required*	The logical library number for which the media information should be reported	integer

Responses

HTTP Code	Description	Schema
200	OK	< V1MediaInfoData > array
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficent rights	No Content
500	Error occurred at operation	Error

Produces

- `application/json`

Security

Type	Name
apiKey	**UserSecurity**

Example HTTP response

Response 200

```
[ {
  "Barcode" : "TC070ML5",
  "LocationType" : "SLOT",
  "LogicalNumber" : "2.34",
  "PhysicalNumber" : 74,
  "Cleaning" : false,
  "LogicalLibrary" : 1,
  "Generation" : 5,
  "SubType" : 0,
  "Protection" : false,
  "Encryption" : false,
  "NoLoads" : 148,
  "MBRead" : 153,
  "MBReadLoad" : 150,
  "MBWritten" : 226,
  "MBWrittenLoad" : 226
}, {
  "Barcode" : "000113L5",
  "LocationType" : "SLOT",
  "LogicalNumber" : "3.4",
  "PhysicalNumber" : 84,
  "Cleaning" : false,
  "LogicalLibrary" : 1,
  "Generation" : 5,
  "SubType" : 0,
  "Protection" : false,
  "Encryption" : false,
  "NoLoads" : 11,
  "MBRead" : 0,
  "MBReadLoad" : 0,
  "MBWritten" : 14,
  "MBWrittenLoad" : 14
} ]
```


2.15. V1-Reports

Reports requests

2.15.1. Get Mount History

```
GET /v1/reports/mountHistory
```

Description

Retrieve a single resource for every drive mount done in the library for the past describing the activity done during a mount. These metrics are updated once a cartridge is unmounted from a drive. This history goes back one year.

Responses

HTTP Code	Description	Schema
200	OK	V1MountHistory
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficent rights	No Content
500	Error occurred at operation	Error

Produces

- `application/json`

Security

Type	Name
apiKey	**AdminSecurity**

Example HTTP response

Response 200

```
[ {
  "barcode" : "000126L7",
  "location" : "drive_M1R2",
  "logicalLibrary" : "",
  "mountTime" : "2020-03-18T13:39:32+0100",
  "unmountTime" : "2020-03-18T13:39:56+0100",
  "hostIOReads" : 500,
  "hostIOWrites" : 12,
  "compressionRate" : 50,
  "errorsCorrectedWrites" : 0,
  "errorsUncorrectedWrites" : 0,
  "errorsCorrectedReads" : 0,
  "errorsUncorrectedReads" : 0
}, {
  "barcode" : "000129L7",
  "location" : "drive_M1R2",
  "logicalLibrary" : "",
  "mountTime" : "2020-03-18T11:37:58+0100",
  "unmountTime" : "2020-03-18T11:38:22+0100",
  "hostIOReads" : 800,
  "hostIOWrites" : 200,
  "compressionRate" : 0,
  "errorsCorrectedWrites" : 0,
  "errorsUncorrectedWrites" : 0,
  "errorsCorrectedReads" : 0,
  "errorsUncorrectedReads" : 0
} ]
```


2.16. V1-System

System requests

2.16.1. Get the current REST API version

```
GET /v1/apiversion
```

Description

The REST API version used in this software release

Responses

HTTP Code	Description	Schema
200	OK	Response 200

HTTP Code	Description	Schema
500	Error occurred at operation	Error

Response 200

Name	Schema
version *required*	string

Example HTTP response

Response 200

```
{
  "version" : "1.1.19"
}
```

Chapter 3. Definitions

3.1. BaseInfoData

The definition of the physical base module information

Name	Description	Schema
SerialNumber *optional*	The serial number for the complete library/stack which is also reported through SCSI	string
MacAdress_1 *optional*	The MAC address of the Ethernet port 1	string
MacAdress_2 *optional*	The MAC address of the Ethernet port 2	string
Vendor *optional*	The product vendor name	string
ProductID *optional*	The Product ID	string
BaseFWRevision *optional*	Firmware Revision of the base library	string
BaseFWBuildDate *optional*	Build date of base library firmware YYYY-MM-DD	string (date)
ExpansionFWRevision *optional*	Firmware Revision of the expansion modules	string
WWNodeName *optional*	World Wide Node Name of the library	string
RoboticHWRevision *optional*	Hardware Revision Level of the Robotic	string

Name	Description	Schema
RoboticFWRevision *optional*	Firmware Revision Level of the Robotic	string
RoboticSerialNumber *optional*	Internal serial number of the Robotic	string
NoOfModules *optional*	Number of connected expansion modules	integer
LibraryType *optional*	Library Type (32/40) (Product specific)	integer

3.2. BaseStatusData

Overview status data of the library

Name	Description	Schema
Information *optional*	Activity Status of the library	string
RobStatus *optional*	Activity Status of the Robotic	string
MoveCount *optional*	Number of moves over lifetime	integer
PowerUpCount *optional*	Number of power cycles over lifetime	integer
PowerOnTime *optional*	Power up time over lifetime	string
LibHealth *optional*	Health Status of library (Summary)	string

3.3. DriveInfo

Name	Description	Schema
LogicalNumber *optional*	The logical drive number starts always at 1 with the first inserted drive in the lowest available module	integer
PhysicalNumber *required*	The physical drive number starting at 1. Formula is (ModuleNo - 1) * (total Drive Slots in single Module) + (DriveSlotNumber in Module)	integer
Module *optional*	The Expansion Module where the drive is located in	integer
Partition *optional*	The partition (logical library) where the drive is assigned to. If no partition is assigned the value will be 0	integer
Generation *optional*	LTO Generation of Drive	integer
Cartridge *optional*	TRUE - Cartridge is loaded / FALSE - no Cartridge loaded	enum (TRUE, FALSE)
Barcode *optional*	The Barcode Label of the cartridge which is loaded in the drive	string
Vendor *optional*	The drive vendor name	string
Product *optional*	The Product ID of the drive	string
FWRevision *optional*	The FW revision of the drive	string
SerialNumber *optional*	The serial number of the drive (Spoofed)	string
WWNodeName *optional*	The World Wide Node Name of the drive	string
Interface *optional*	Interface Type - SAS / Fibre Channel	string

Name	Description	Schema
MFGSerialNumber *optional*	The Manufacturing Serial number	string
ErrorState *optional*	Is Drive in an error state	enum (TRUE, FALSE)
Power *optional*	Is drive powered	enum (TRUE, FALSE)
Presence *optional*	Is drive present	enum (TRUE, FALSE)
ADTMode *optional*	ADT Transport Mode	string

3.4. DriveInfoInventory

Name	Description	Schema
PhysicalNumber *required*	The physical drive number starting at 1. Formula is (ModuleNo - 1) * (total Drive Slots in single Module) + (DriveSlotNumber in Module)	integer
LogicalNumber *optional*	The logical drive number starts always at 1 with the first inserted drive in the lowest available module	integer
Module *optional*	The Expansion Module where the drive is located in	integer
Partition *optional*	The partition (logical library) where the drive is assigned to. If no partition is assigned the value will be 0	integer
Barcode *optional*	The Barcode Label of the cartridge which is loaded in the drive	string
Vendor *optional*	The drive vendor name	string
Product *optional*	The Product ID of the drive	string

Name	Description	Schema
FWRevision *optional*	The FW revision of the drive	string
SerialNumber *optional*	The serial number of the drive (Spoofed)	string
MFGSerialNumber *optional*	The manufacturing serial number	string

3.5. Error

Name	Schema
code *optional*	string
message *optional*	string

3.6. EventEntry

Structure of event entry

Name	Description	Schema
Number *required*	Number of event in list	integer
TimeStamp *optional*	Date/ Time of of event	string
EventCode *optional*	Eventcode, describing the event which occurred	integer
Text *optional*	Short description of EventCode	string
State *optional*	Shows the current state for the ticket.	enum (INFO, CONFIG, SERVICE, TICKET_PENDING, TICKET_RESOLVED, SINGULAR_TICKET)

Name	Description	Schema
Closed *optional*	Reflects the closed state for tickets	enum (TRUE, FALSE)

3.7. IOStatus

IO Station (Mailslot) status data

Type : < IOStatus > array

IOStatus

Name	Description	Schema
ModuleNo *required*	logical module number where the IO Station is located	integer
Configured *optional*	FALSE - IO Station Slots are configured as storage slots. TRUE - IO Station Slots are configured fo r IO Operation	enum (TRUE, FALSE)
Unlocked *optional*	FALSE - IO Station is locked. TRUE - IO Station is unlocked	enum (TRUE, FALSE)
OpenStatus *optional*	FALSE - IO STation is closed. TRUE - IO Station is opened	enum (TRUE, FALSE)

3.8. Inventory

Name	Schema
Slots *optional*	< Slot > array
Drives *optional*	< DriveInfoInventory > array

3.9. LibraryInfo

The definition of the physical library information

Name	Description	Schema
BaseInfo *required*		BaseInfoData

Name	Description	Schema
ModulesInfo *required*	Module Information	< Module > array

3.10. LibraryStatus

Overview status information of the library

Name	Schema
BaseStatus *required*	BaseStatusData
ModulesStatus *required*	< ModuleStatusData > array

3.11. LicenseInfo

Licensed Feature data

Name	Description	Schema
Feature *required*	Name of licensed feature	string
LicenseKey *required*	License Key string	string

3.12. MediaInfoData

Information about a tape media.

Name	Description	Schema
Barcode *optional*	The Barcode Label of the media	string
LocationType *optional*	Type of location where the media is stored. "Drive" or "Slot"	enum (DRIVE, SLOT)
LogicalNumber *optional*	The logical number of media location	string

Name	Description	Schema
PhysicalNumber *optional*	The physical number of media location	string
Cleaning *optional*	TRUE - media is a cleaning tape / FALSE - media is not a cleaning tape	enum (FALSE, TRUE)
Partition *optional*	The partition (logical library) where the media is assigned to. If no partition is assigned the value will be 0	integer
Generation *optional*	LTO Generation of media	integer
SubType *optional*	LTO SubType of media (0 = default, 1= Type M)	integer
Protection *optional*	TRUE - media is write protected / FALSE - media is not protected	enum (FALSE, TRUE)
Encryption *optional*	TRUE - media is encrypted / FALSE - media is not encrypted / Unknown if not identified by drive	enum (FALSE, TRUE, Unknown)
NoLoads *optional*	Number of loads of media. Shows how many times the media was loaded into a tape drive	integer
MBRead *optional*	Bytes read from this media in MB	integer
MBReadLoad *optional*	Bytes read from this media in MB during last load	integer
MBWritten *optional*	Bytes written to this media in MB	integer
MBWrittenLoad *optional*	Bytes written to this media in MB during last load	integer

3.13. Module

The definition of an expansion module informational data

Name	Description	Schema
PhysicalNumber *required*	The physical number of the expansion module. the numbering starts always at 1 with the bottom module counting to 7 regardless of the physical presence of this module. The base module has always the number 4	integer
LogicalNumber *optional*	The number of the module in an existing library stack. The lowest module always starts with 1	integer
ReadyStatus *optional*	This entry returns the current status of the module. If it is correctly initialized and running it will be set to TRUE	enum (TRUE, FALSE)
SerialNumber *optional*	The internal manufacturing serial number of the module	string

3.14. ModuleStatusData

Current Health Status of a module

Name	Description	Schema
PhysicalNumber *required*	Module Number (physically)	integer
LogicalNumber *optional*	Module Number (logical)	integer
Health *optional*	Current health status of Module	string

3.15. Modules

List of expansion module data

Type : < Module > array

3.16. PartitionInfo

Partition data

Name	Description	Schema
PartitionNumber *required*	Number of partition	integer
Name *optional*	Name of partition	string
SerialNumber *optional*	Serialnumber of partition, based on serialnumber of physical library	string
NumSlots *optional*	Number of slots assigned to partition	integer
NumIOSlots *optional*	Number of IO slots (Mailslots) assigned to partition	integer
NumDrives *optional*	Number of drives assigned to partition	integer
LunMasterDrive *optional*	Lun Master Drive number (logical)	integer
LunMasterDrivePhys *optional*	Lun Master Drive number (physical)	integer
LunMasterDriveArr *optional*	Lun Master Drive number (logical)	< integer > array
LunMasterDrivePhysArr *optional*	Lun Master Drive number (physical)	< integer > array

Name	Description	Schema
EncryptionMode *optional*	Encryption Mode which is currently active for this partition. The following values are possible • OFF - No encryption policy activated • ISV - Encryption can be controlled by host application • KMIP - KMIP Key Management is activated. A KMIP Key Server is required • ESKM - ESKM Key Management is activated. A ESKM Server is required • PLK - Local Key Management over a PLK Token is active. A PLK token is required	enum (OFF, ISV, KMIP, ESKM, PLK, IPP)
BarcodeAlign *optional*	Barcode Label Alignment (left,right)	enum (left, right)
BarcodeLength *optional*	Barcode Label Length	integer
AutoClean *optional*	AutoClean enabled	enum (TRUE, FALSE)
WWNode *optional*	WWN of Partition	string
Micw *optional*	Multi-initiator conflict warning enabled	enum (TRUE, FALSE)

3.17. Slot

The definition of cartridge slot data

Name	Description	Schema
PhysicalNumber *required*	The physical slot number starting at 1. Formula is (ModuleNo - 1) * (total Slots in single Module) + (SlotNumber in Module)	integer
LogicalNumber *optional*	The logical slot number includes the module number as prefix and starts always at 1 in a module. Format - ModuleNo.(SlotNumber in Module)	string

Name	Description	Schema
Module *optional*	The Expansion Module where the slot is located in	integer
Partition *optional*	The partition (logical library) where the slot is assigned to. If no partition is assigned the value will be 0	integer
Mailslot *optional*	If the slot is mapped as a mailslot (I/O Station) this entry is set to TRUE	enum (TRUE, FALSE)
Cartridge *optional*	If a cartridge is detected in the slot this entry is set to TRUE	enum (TRUE, FALSE)
CartridgeType *optional*	Type of cartridge Data,Cleaning,Worm etc.	enum (Unknown, Data, Cleaning, Worm, N/A)
CartridgeSubType *optional*	SubType of a cartridge 0 = Default, 1= Type M	integer
CartridgeGeneration *optional*	Generation of the cartridge	integer
CartridgeEncrypted *optional*	Cartridge encryption state	enum (TRUE, FALSE, Unknown)
Barcode *optional*	The Barcode Label of the the cartridge which is sitting in the slot	string
Access *optional*	If the slot is accessible by the system the entry is set to TRUE	enum (TRUE, FALSE)
Blocked *optional*	TRUE - The slot is not reachable by the robot / FALSE - Slot is reachable by the robot **Default** : `"FALSE"`	enum (TRUE, FALSE)

3.18. TestStatus

Status of current running test

Name	Description	Schema
currentCycle *optional*	current test cycle number	integer
testCycles *optional*	number of test cycles planned	integer
status *optional*	status of the test like: not started, running, passed, failed or stopped by user request.	string
running *optional*	TRUE - test is still running, FALSE - no test running	enum (TRUE, FALSE)

3.19. V1Accessor

Name	Description	Schema
location *optional*	"Always returns a value of 'accessor_Aa'."	string
state *optional*	A string representing the current state of the accessor. Values include the following in priority order: failed, online	string
moves *optional*	An integer count of the number of moves this accessor has performed in its lifetime.	integer

3.20. V1DriveInfo

Name	Description	Schema
location *optional*	A string representing the unique location of the tape drive. This is also the unique identifier for the drive. The format is "drive_M<m>R<r>" where ‘m’ is the module and ‘r’ is the row or module location (1 through 3 within the module) the drive is installed in.	string
sn *optional*	A string representing the spoofed serial number of the drive. This can also be used as a unique identifier for the tape drive.	string
mfgSn *optional*	A string representing the manufacturing serial number of the drive. This can also be used as a unique identifier for the tape drive.	string

Name	Description	Schema
mediaType *optional*	A string representing the media type supported by the tape drive. This always returns "LTO".	string
state *optional*	A string representing the current state of the tape drive. Values include the following in priority order: • online o Logged in and empty according to library inventory • cleaning o Not empty and status reported by drive shows activity • loading • unloading • reading • writing • locating • rewinding • erasing • formatting • calibrating • updating • ready o Status reported by drive shows no activity but mounted. NOTE: This will normally match the drive's SCSI ready condition but not always. • unloaded o Status reported by drive shows no activity and not mounted	string
mtm *optional*	A string representing the machine type and model of the tape drive (e.g. ULT3580-HH8)	string
interface *optional*	A string representing the type of ports this tape drive includes. Values include: • fibreChannel • SAS	string
logicalLibrary *optional*	A string representing the name of the logical library this tape drive has been assigned to or "null" if the tape drive is currently unassigned.	string
use *optional*	A string representing the assigned use this tape drive has been given within the logical library. Values include: • "access" o data access drives • "controlPath" o data access and control path drives	string
firmware *optional*	A string representing the firmware version currently installed on the tape drive.	string
encryption *optional*	The state of encryption on this tape drive. Values include: "enabled" and "disabled".	string
mounts *optional*	The total count of mounts.	integer
barcode *optional*	A string representing the SN barcode label found on the tape drive.	string
wwnn *optional*	A 16-character hex string representing the world-wide node name of the tape drive.	string

Name	Description	Schema
elementAddress *optional*	An integer representing the element address of the tape drive.	integer

3.21. V1DriveInfoNaming

Name	Description	Schema
LogicalNumber *optional*	The logical drive number starts always at 1 with the first inserted drive in the lowest available module	integer
PhysicalNumber *required*	The physical drive number starting at 1. Formula is (ModuleNo - 1) * (total Drive Slots in single Module) + (DriveSlotNumber in Module)	integer
Module *optional*	The Expansion Module where the drive is located in	integer
LogicalLibrary *optional*	The logical library where the drive is assigned to. If no logical library is assigned the value will be 0	integer
Generation *optional*	LTO Generation of Drive	integer
Cartridge *optional*	TRUE - Cartridge is loaded / FALSE - no Cartridge loaded	boolean
Barcode *optional*	The Barcode Label of the cartridge which is loaded in the drive	string
Vendor *optional*	The drive vendor name	string
Product *optional*	The Product ID of the drive	string
FWRevision *optional*	The FW revision of the drive	string
SerialNumber *optional*	The serial number of the drive (Spoofed)	string

Name	Description	Schema
WWNodeName *optional*	The World Wide Node Name of the drive	string
Interface *optional*	Interface Type - SAS / Fibre Channel	string
MFGSerialNumber *optional*	The Manufacturing Serial number	string
ErrorState *optional*	Is Drive in an error state	boolean
Power *optional*	Is drive powered	boolean
Presence *optional*	Is drive present	boolean
ADTMode *optional*	ADT Transport Mode	string

3.22. V1DriveInventory

Name	Description	Schema
LogicalNumber *optional*	The logical drive number starts always at 1 with the first inserted drive in the lowest available module	integer
PhysicalNumber *required*	The physical drive number starting at 1. Formula is (ModuleNo - 1) * (total Drive Slots in single Module) + (DriveSlotNumber in Module)	integer
Module *optional*	The Expansion Module where the drive is located in	integer
LogicalLibrary *optional*	The logical library where the drive is assigned to. If no logical library is assigned the value will be 0	integer

Name	Description	Schema
Barcode *optional*	The Barcode Label of the cartridge which is loaded in the drive	string
Vendor *optional*	The drive vendor name	string
Product *optional*	The Product ID of the drive	string
FWRevision *optional*	The FW revision of the drive	string
SerialNumber *optional*	The serial number of the drive (Spoofed)	string

3.23. V1EthernetPort

Name	Description	Schema
location *optional*	A string representing the unique location of this port. The format of this field is "ethernetPort_M<m>P<p>" where <m> is the module and <p> is the port number.	string
macAddress *optional*	The media access control (MAC) address of this ethernet port.	string
ipv4Address *optional*	The IPv4 address of this ethernet port.	string
ipv4Subnet *optional*	The IPv4 subnet mask of this ethernet port.	string
ipv4Gateway *optional*	The IPv4 gateway address of this ethernet port.	string
ipv4Assignment *optional*	The IPv4 address of this ethernet port. Values include: • "static" • "dynamic"	string
ipv4Primary *optional*	The IPv4 primary DNS address of this ethernet port. If the ipv4Assignment is "Static", this is null.	string

Name	Description	Schema
ipv6Address *optional*	The IPv6 address of this ethernet port.	string
ipv6PrefixLength *optional*	The IPv6 prefix length of this ethernet port.	string
ipv6Gateway *optional*	The IPv6 gateway address of this ethernet port.	string
ipv6Primary *optional*	The IPv6 primary DNS address of this ethernet port.	string
ipv6Link *optional*	The IPv6 link address of this ethernet port.	string
ipv6DHCP *optional*	The IPv6 DHCP setting of this ethernet port.	string
ipv6StatelessConfig *optional*	The IPv6 stateless config setting of this ethernet port.	string
ipv6Static *optional*	The IPv6 static address of this ethernet port.	string

3.24. V1EventEntry

Structure of event entry

Name	Description	Schema
ID *optional*	A string that is the unique identifier of the event in the library. EVENTPrefix format : • C -> Configuration • I -> Information • T -> Ticket like (EVENTPrefix+ID like C338, I123, T113).	string
severity *optional*	The severity of the event which represents the urgency that should be given to it. Values include: • error • warning • inactiveError • inactiveWarning • information • configuration	string

Name	Description	Schema
time *optional*	The time the event occurred. Time format is in ISO 8601 format of "YYYY-MM-DDThh:mm:ss+-hhmm". For example, "2018-09-17T23:02:00-0700".	string
type *optional*	Event type	string
location *optional*	A string representing the hardware component where the event originated. This can be "null" if no hardware was associated with this event. Hardware components include: • accessor_Aa • module_M<m> (e.g. "module_M1") • drive_M<m>R<r> (e.g. "drive_M3R2")	string
description *optional*	A string describing the event that gives more information on exactly what it represents. This includes the list of parameters unique to this event type. For example, with the event type of "A user logged in at the Web GUI Interface", it would show a description of: "A user logged in at the Web GUI interface (SYS_COMPONENT: SYSTEM, USER_NAME: administrator, USER_ROLE: Administrator, USER_REMOTE_IP: 9.24.29.33)."	string
errorCode *optional*	A string that represents the internal error code used to represent this type of event. For example "4072:, "4019", etc.	string

3.25. V1Inventory

Name	Schema
Slots *optional*	< V1SlotInventory > array
Drives *optional*	< V1DriveInventory > array

3.26. V1LibraryEntry

Name	Description	Schema
name *optional*	A string representing the name the tape library was given. This always returns null.	string

Name	Description	Schema
status *optional*	A string representing the overall health status of the library. Values are: • ERROR • WARNING • OK	string
totalCapacity *optional*	An integer representing the total physical cartridge capacity of the library.	integer
licensedCapacity *optional*	An integer representing the total licensed cartridge capacity of the library.	integer
totalCartridges *optional*	An integer representing the total number of cartridges in the library slots, I/O stations, drives, and accessors currently.	integer
assignedCartridges *optional*	An integer representing the total number of cartridges assigned to logical libraries in the library.	integer
firmware *optional*	A string representing the firmware level installed on the library.	string
sn *optional*	A string representing the serial number of the library.	string
time *optional*	The current date and time set on the library. All dates and times returned by the library are in relation to this time. Time format is in ISO 8601 format of "YYYY-MM-DDThh:mm:ss+-hhmm". For example, "2018-09-17T23:02:00-0700".	string
secureCommunications *optional*	The current setting for secure communications. If "enabled", the user is required to log into the GUI using a secure HTTPS connection. If "disabled" an unsecured HTTP connection is allowed.	enum (enabled, disabled)

3.27. V1LogicalLibraryInfo

Logical Library data

Name	Description	Schema
LogicalLibraryNumber *required*	Number of logical library	integer
Name *optional*	Name of logical library	string
SerialNumber *optional*	Serialnumber of logical library, based on serialnumber of physical library	string
NumSlots *optional*	Number of slots assigned to logical library	integer
NumIOSlots *optional*	Number of IO slots assigned to logical library	integer
NumDrives *optional*	Number of drives assigned to logical library	integer
LunPrimaryDrive *optional*	Lun Primary Drive number (logical)	integer
LunPrimaryDrivePhys *optional*	Lun Primary Drive number (physical)	integer
LunPrimaryDriveArr *optional*	Lun Primary Drive number (logical)	< integer > array
LunPrimaryDrivePhysArr *optional*	Lun Primary Drive number (physical)	< integer > array

Name	Description	Schema
EncryptionMode *optional*	Encryption Mode which is currently active for this logical library. The following values are possible • OFF - No encryption policy activated • ISV - Encryption can be controlled by host application • KMIP - KMIP Key Management is activated. A KMIP Key Server is required • ESKM - ESKM Key Management is activated. A ESKM Server is required • PLK - Local Key Management over a PLK Token is active. A PLK token is required	enum (OFF, ISV, KMIP, ESKM, PLK, IPP)
BarcodeAlign *optional*	Barcode Label Alignment (left,right)	enum (left, right)
BarcodeLength *optional*	Barcode Label Length	integer
AutoClean *optional*	AutoClean enabled	boolean
WWNode *optional*	WWN of Partition	string
Micw *optional*	Multi-initiator conflict warning enabled	boolean

3.28. V1MediaInfoData

Information about a tape media.

Name	Description	Schema
Barcode *optional*	The Barcode Label of the media	string
LocationType *optional*	Type of location where the media is stored. "Drive" or "Slot"	enum (DRIVE, SLOT)

Name	Description	Schema
LogicalNumber *optional*	The logical number of media location	string
PhysicalNumber *optional*	The physical number of media location	string
Cleaning *optional*	TRUE - media is a cleaning tape / FALSE - media is not a cleaning tape	boolean
LogicalLibrary *optional*	The logical library where the media is assigned to. If no partition is assigned the value will be 0	integer
Generation *optional*	LTO Generation of media	integer
SubType *optional*	LTO SubType of media (0 = default, 1= Type M)	integer
Protection *optional*	TRUE - media is write protected / FALSE - media is not protected	boolean
Encryption *optional*	TRUE - media is encrypted / FALSE - media is not encrypted	boolean
NoLoads *optional*	Number of loads of media. Shows how many times the media was loaded into a tape drive	integer
MBRead *optional*	Bytes read from this media in MB	integer
MBReadLoad *optional*	Bytes read from this media in MB during last load	integer
MBWritten *optional*	Bytes written to this media in MB	integer
MBWrittenLoad *optional*	Bytes written to this media in MB during last load	integer

3.29. V1MountHistory

Name	Description	Schema
barcode *optional*	A string representing the volume serial number or barcode that uniquely identifies the cartridge to a host. For example,"CLNU28L1"or "null" if the VOLSER is unknown	string
location *optional*	A string representing the unique location of the tape drive where the mount occurred. The format is "drive_M<m>R<r>" where ‘m’ is the module and ‘r’ is the row or module location (1 through 3 within the module) the drive is installed in.	string
logicalLibrary *optional*	The unique name of the logical library the cartridge and drive belong to.	string
mountTime *optional*	The current date and time the mount occurred. Time format is in ISO 8601 format of “YYYY-MM-DDThh:mm:ss+-hhmm”. For example, “2018-09-17T23:02:00-0700”.	string
unmountTime *optional*	The current date and time the cartridge was unmounted. Time format is in ISO 8601 format of “YYYY-MM-DDThh:mm:ss+-hhmm”. For example, “2018-09-17T23:02:00-0700”.	string
hostIOReads *optional*	The total number of MB of data read from the cartridge on this mount, not counting compression and formatting overhead.	integer
hostIOWrites *optional*	The total number of MB of data written to the cartridge on this mount, not counting compression and formatting overhead.	integer
compressionRate *optional*	Compression rate percentage	integer
errorsCorrectedWrites *optional*	An integer representing the number of corrected write errors that occurred during the mount.	integer
errorsUncorrectedWrites *optional*	An integer representing the number of uncorrected write errors that occurred during the mount.	integer

Name	Description	Schema
errorsCorrectedReads *optional*	An integer representing the number of corrected read errors that occurred during the mount.	integer
errorsUncorrectedReads *optional*	An integer representing the number of uncorrected read errors that occurred during the mount.	integer

3.30. V1SlotInventory

The definition of cartridge slot data

Name	Description	Schema
PhysicalNumber *required*	The physical slot number starting at 1. Formula is (ModuleNo - 1) * (total Slots in single Module) + (SlotNumber in Module)	integer
LogicalNumber *optional*	The logical slot number includes the module number as prefix and starts always at 1 in a module. Format - ModuleNo.(SlotNumber in Module)	string
Module *optional*	The Expansion Module where the slot is located in	integer
LogicalLibrary *optional*	The logical library where the slot is assigned to. If no logical library is assigned the value will be 0	integer
Mailslot *optional*	If the slot is mapped as a mailslot (I/O Station) this entry is set to TRUE	boolean
Cartridge *optional*	If a cartridge is detected in the slot this entry is set to TRUE	boolean
CartridgeType *optional*	Type of cartridge Data,Cleaning,Worm etc.	enum (Unknown, Data, Cleaning, Worm, N/A)
CartridgeSubType *optional*	SubType of a cartridge 0 = Default, 1= Type M	integer

Name	Description	Schema
CartridgeGeneration *optional*	Generation of the cartridge	integer
CartridgeEncrypted *optional*	Cartridge encryption state	boolean
Barcode *optional*	The Barcode Label of the the cartridge which is sitting in the slot	string
Access *optional*	If the slot is accessible by the system the entry is set to TRUE	boolean
Blocked *optional*	TRUE - The slot is not reachable by the robot / FALSE - Slot is reachable by the robot	boolean

Chapter 4. Security

4.1. UserSecurity

User role API Key, only allowing status requests.

Type : apiKey
Name : Authorization
In : HEADER

4.2. AdminSecurity

Admin role API Key, allowing status, Operations and operational requests

Type : apiKey
Name : Authorization
In : HEADER

4.3. ServiceSecurity

Service role API Key, allowing special service requests including Admin level

Type : apiKey
Name : Authorization
In : HEADER