



# Intel Accelerated Storage Manager

VROC Plugin REST API

**INTEL CONFIDENTIAL**

*November 2018*

*Document Revision 2.0.1*

## Contents

Revision History.....	3
<b>1. Introduction .....</b>	<b>6</b>
<b>2. URI structure .....</b>	<b>6</b>
<b>3. Response handling .....</b>	<b>6</b>
<b>4. OAuth parameters.....</b>	<b>7</b>
<b>5. Resources.....</b>	<b>7</b>
5.1 VROC state.....	7
5.2 Version information .....	8
5.3 Configuration information.....	9
5.4 List of RAIDs.....	10
5.5 List of controllers associated with RAID.....	18
5.6 Controllers information .....	22
5.7 End device information associated with controller .....	25
5.8 List of arrays associated with controller .....	27
5.9 Ports information associated with controller.....	28
5.10 List of volumes associated with controller .....	29
5.11 Ports information .....	31
5.12 End device information.....	33
5.13 List of all arrays.....	35
5.14 End device information associated with array .....	36
5.15 List of volumes associated with array .....	39
5.16 List of all volumes.....	41
5.17 Events information.....	43
<b>6. Actions .....</b>	<b>44</b>
6.1 Edit configuration.....	44
6.2 Clear metadata.....	45
6.3 Mark/unmark disk as spare .....	45
6.4 Mark disk as normal.....	46
6.5 Reset disk SMART event .....	47
6.6 Managed Hotplug .....	48
6.7 Mark volume as normal.....	49
6.8 Rebuild volume.....	49
6.9 Delete volume .....	50
6.10 Create volume .....	51
6.11 Modify volume name .....	55
6.12 Modify volume size .....	56
6.13 Modify volume cache policy.....	57
6.14 Modify volume RAID level .....	58
6.15 Modify volume RAID Write Hole .....	59
6.16 Initialize volume .....	60
6.17 Start/Cancel volume verification.....	61
6.18 Add disks to array.....	62
6.19 Set array write cache state .....	63
6.20 Port locate .....	64
6.21 Rescan for hardware changes .....	65
6.22 Read patrol .....	65
6.23 Rebuild on hot insert .....	66

## Revision History

Revision	Contributor	Description	Date
0.1	Marek Bielańczuk	Initial version of document.	March 2015
0.1.1 - 0.1.9	Team	Peer editing	March-December 2015
0.2	Marek Bielańczuk	Merge and cleanup	December 2015
0.2.1	Marek Bielańczuk	Added error IDs for requests responses.	December 2015
0.2.2	Marek Bielańczuk	Added parameter "repair" for verify action.	December 2015
0.2.3	Marek Bielańczuk	Added get state request (added "objectXxx" parameters and "BUSY" state).	December 2015
0.2.4	Marek Bielańczuk	Removed "arrayId" from "Mark/Unmark disk as spare". Removed "storagePool" and added "PCISlotNumber" to end device.	December 2015
0.2.5	Marek Bielańczuk	Added events list.	December 2015
0.2.6	Piotr Wróblewski	Modify volume changes.	December 2015
0.2.7	Piotr Wróblewski	Removed <i>diskUnlock</i> from controller. Create volume size description. Updated volume cache policy.	January 2016
0.2.8	Piotr Wróblewski	Modify read patrol state.	January 2016
0.2.9	Piotr Wróblewski	Volume strip size details.	January 2016
0.2.10	Piotr Wróblewski	Volume name changes. Modify volume size for RAID_0 is not supported. OAuth parameters. Volume verify repair action changed.	January 2016
0.2.11	Marek Bielańczuk	Modify volume size type changed to string.	February 2016
0.2.12	Piotr Wróblewski	Added request example to Mark/unmark disk as spare. Modify volume changes.	February 2016
0.3	Piotr Wróblewski	Combined RAID information and RAID levels. Added new Contents headers. Added INTERNAL_ERROR error code to returned error IDs. Added "x8Disk" to enddevices for x8 support. Added RSTe_INVALID_RAID_LEVEL error code for create and modify volume. Added NOT_SUPPORTED error code for adding disk to array.	March 2016
0.3.1	Piotr Wróblewski	Added RSTe_UNSUPPORTED_RAID_LEVEL and INVALID_PARAMETER error codes to Raids.	March 2016
0.4.0	Piotr Wróblewski	Added support for "rwhPolicy".and "journalingDriveld" for Volumes under controllers, arrays, volumes and create volume. Added JOURNALING_DRIVE type to "usage" field in end devices. Added error code NOT_SUPPORTED for modify RAID level and create volume. Added error code RSTe_SUCCESS_NO_RWH for create volume. Added new action Modify volume RAID Write Hole. Changed VOLUME_DEGRADATED to VOLUME_DEGRADED.	March 2016
0.4.1	Piotr Wróblewski	Added support for VMD. Added "vmdDomain" field to enddevices under controllers, arrays and enddevices.  Changed VERIFYING_AND_FIX to VERIFYING_AND_FIXING.	March 2016
0.4.2	Piotr Wróblewski	Removed "supported" field from RAID level information.	March 2016

0.5	Piotr Wróblewski	Modify volume name, cache policy and size as separate actions. Added rebuild on hot insert.	April 2016
0.6	Piotr Wróblewski	Added responses to actions.	April 2016
0.6.1	Piotr Wróblewski	Removed unused <i>spanned</i> field from list of arrays.	April 2016
0.6.2	Piotr Wróblewski	Added rebuild on hot insert options to controllers information.	April 2016
0.6.3	Piotr Wroblewski	Added <i>availableDevices</i> to controllers information.	April 2016
0.6.4	Piotr Wróblewski	Added rebuild volume example request. Fixed set array write cache state to array response.	April 2016
0.6.5	Piotr Wróblewski	Added create volume example with RWH. Fixed <i>EndDevices blockSize</i> description. Added <i>diskId</i> NOT_SUPPORTED in create volume. Removed <i>rwhPolicy</i> and <i>journalingDriveld</i> from modify volume RAID level.	April 2016
0.6.6	Piotr Wróblewski	Restored RAID level <i>stripSizesSupported</i> field. Fixed example responses for RAIDs. Optional <i>stripSize</i> in modify volume RAID level.	April 2016
0.6.7	Piotr Wróblewski	Removed optional for <i>rwhPolicy</i> in modify volume RWH request and response. Added optional to <i>stripSize</i> and <i>disks</i> in create volume. Removed <i>diskUnlock</i> and <i>assignStoragePool</i> from list of controllers associated with RAID. Changed controller <i>vendorId</i> and <i>subVendorId</i> type to string. Changed state values for action responses. Removed <i>ledState</i> , <i>PCISlotNumber</i> and <i>systemIOBusNumber</i> from <i>enddevices</i> .	May 2016
0.6.8	Marek Bielańczuk	Updated elements: - objects ID type: - 5.1 – <i>inProgress</i> property description, - 5.3.7 – Removed additional raid object from example, - 5.4 – updated JSON example - 5.5 – updated id, raid and <i>blocksFree</i> - 5.9 – <i>rwhPolicy</i> as Optional, updated JSON example, - 5.11 – <i>raidId</i> property description, - 5.13 – <i>raidId</i> property description, - 5.14 – <i>rwhPolicy</i> as Optional, updated JSON example, - 5.15 – <i>rwhPolicy</i> as Optional, updated JSON example, - 6.2 – action property description, - 6.4 – state property return enum value, - 6.7 – curl example, <i>disks</i> property, JSON example, - 6.9 – curl example, JSON example - 6.10 – curl example, JSON example, - 6.11 – JSON example, - 6.12 – curl example - 6.14 – request description, - 6.15 – curl example, - 6.16 – state property, <i>disks</i> and, <i>writeCachePolicy</i> property, - 6.17 – state description.	August 2016
0.6.9	Marek Bielańczuk	- Page 9: <i>StripSize</i> "N/A" in examples for RAID 1 and 2 - Page 11: <i>raidId</i> to {String} - Page 15: <i>id</i> {String} Controller Identifier. - Page 17: <i>controllerId</i> and <i>raidId</i> to {String} - Page 22: <i>journalingDriveld</i> to {String} - Page 28: <i>errors</i> and <i>badBlocks</i> to {Number}, <i>journalingDriveld</i> to {String} <i>raidLevel</i> example to RAID 5 - Page 29: <i>errors</i> and <i>badBlocks</i> to {Number}, <i>journalingDriveld</i> to {String} <i>raidLevel</i> example to RAID 5 - Page 37: <i>journalingDriveld</i> to {String} - Page 38: <i>errors</i> and <i>badBlocks</i> to {Number}, <i>journalingDriveld</i> to {String} <i>raidLevel</i> example to RAID 5 - Page 43: <i>journalingDriveld</i> to {String} - Page 46: curl example fix for "state" - Page 47: curl example fix for "state" - Page 48: curl example fix for "state" - Page 49: curl example fix for "state"	August 2016

0.7.0	Marcin Dembiński	Added information that /rste/raids shows only supported RAIDs Added hardwareKey and supportsTPV fields for /rste/controllers Added vendorId and isIntelNVMe fields for /rste/enddevices Added additional values for EndDevice status	October 2016
1.0	Marek Bielańczuk	Added controller type (SATA), changed end device 'vendorId' field to optional	November 2016
1.1	Marek Bielańczuk	Removed FD internal name	November 2016
1.2	Marcin Dembiński	Added reset SMART event call information	November 2016
1.3	Marcin Dembiński	Renamed 3_STORY key to VROC_PASS_THRU key Removed N/A and SIZE_UNKNOWN stripSize for Requests	May 2017
1.4	Marcin Dembiński	Added VROC_INTEL_ONLY_SKU hardware key for controller	May 2017
1.4.1	Marcin Dembiński	Added managedHotplug field for /rste/enddevices Added DELETE /rste/disks/:diskId as Managed Hotplug request	May 2017
1.4.2	Marcin Dembiński	Added created field for /rste/raids for RaidLevelInfo Added trial field for /rste/version	May 2017
1.4.3	Marcin Dembiński	Updated information for /rste/enddevices (and for controllers and arrays): Changed blocksFree to freeSize Removed blocksTotal Changed blockSize to sectorSize object	June 2017
1.4.4	Marcin Dembiński	Added missing managedHotplug field for endDevice in controller and array calls	June 2017
1.4.5	Marcin Dembiński	Removed address field from Controller	June 2017
1.4.6	Marcin Dembiński	Restored address field in Controller but only for Linux	June 2017
1.4.7	Marcin Dembiński	Added defaultStripSizes field in RAID Level Info	June 2017
1.4.8	Marcin Dembiński	Removed N/A stripSize	September 2017
1.4.9	Marcin Dembiński	Changed VERIFYING_AND_FIX to VERIFYING_AND_FIXING in missing places	September 2017
1.4.10	Marcin Dembiński	Replaced <i>null</i> with <i>[]</i> in <i>stripSizesSupported</i> field in an example	October 2017
1.4.11	Marcin Dembiński	Remove <i>for RAID 1 stripSize is not present</i> from responses	November 2017
1.4.12	Marcin Dembiński	Added GET and PUT /rste/config requests	January 2018
1.4.13	Marcin Dembiński	Unified RAID levels in documentation to the form: RAID_X instead of the mix of RAIDX, RAID_X and RAID X	January 2018
2.0	Marcin Dembiński	Global refactor of documentation	May 2018
2.0.1	Klaudia Jabłońska	Update description of 'Initialize volume' request – not supported on Linux.	November 2018

## 1. Introduction

This document is a reference for Intel VROC REST API. Scope of this API covers access to VROC plugin resources on host machine where agent is installed.

## 2. URI structure

Access to resources is provided via URI paths. Client applications should send requests over the HTTP protocol using standard GET, POST and DELETE methods.

Intel ASM VROC Plugin URI structure:

```
{protocol}://{host}[:{port}]/v1/vroc/{resource}[?{param}={value}]
```

Where:

<b>{protocol}</b>	Communication protocol – <b>http</b> or <b>https</b>
<b>{host}</b>	Host machine IP address or domain name where IASM is installed
<b>{port}</b>	Port number on which IASM service is listening. Optional if port is default for its protocol ( <b>80</b> for http, <b>443</b> for https)
<b>v1</b>	Stands for first version of API definition
<b>vroc</b>	Namespace for all resources related to VROC Plugin
<b>{resource}</b>	Name of resource to be returned or modified
<b>{parameter}</b>	Name of URI query parameter
<b>{value}</b>	Value of URI query parameter

## 3. Response handling

REST API returns responses in wrapper format presented below. Note: Even response with error returns Status 200 OK.

Response format		
<b>data</b>	object or array	Contains the request response. It can be either an array or an object. Null if <b>success</b> is false.
<b>status</b>	object	Contains the response status object.
<b>{}.success</b>	bool	Response status. Determines if the request was successful or end with an error.
<b>{}.errorId</b>	string	<i>[Optional parameter]</i> Identifier of the return error. Present only if <b>success</b> is false.
<b>{}.errorMessage</b>	string	<i>[Optional parameter]</i> Additional, detailed message which describes occurred error and its reason. Present only if <b>success</b> is false.

## Response examples:

```
# Successful request's response example
Status: 200 OK
{
  "data": {
    # returned information from REST API
  },
  "status": {
    "success": true
  }
}
```

```
# Failed request's response example
Status: 200 OK
{
  "data": null,
  "status": {
    "errorId": "VROC_OFFLINE",
    "errorMessage": "VROC plugin does not work",
    "success": false
  }
}
```

## 4. OAuth parameters

To get access to VROC Plugin resources the OAuth scope parameter must contain value "vroc":

scope=vroc

## 5. Resources

Below is a list of all available resources for VROC plugin.

### 5.1 VROC state

[GET] /v1/vroc/state		
Returns VROC product state and list of currently running actions.		
Request parameters		
<i>none</i>		
Response parameters (Object)		
<b>inProgress</b>	array	Array of currently running actions. Can be empty.
<b>[].action</b>	string	Name of an action Possible values: INITIALIZING, REBUILDING, VERIFYING, VERIFYING_AND_FIXING, GENERAL_MIGRATION.
<b>[].objectID</b>	string	ID of the object on which action is currently running.
<b>[].objectName</b>	string	Name of the object.
<b>[].objectType</b>	string	Type of the object. Possible values: VOLUME.

<b>state</b>	string	VROC plugin internal state. Possible values: NORMAL, WARNING, FAILED.
<b>Available Errors</b>		
<b>INVALID_PARAMETER</b>		Additional field's been attached with the request.
<b>VROC_OFFLINE</b>		VROC Plugin is not working (lack of a driver or an incompatible version).
<b>INTERNAL_ERROR</b>		An unexpected error occurred during request process.

## Request Example:

```
GET https://example.com/v1/vroc/state
```

```
{
  "data": {
    "InProgress": [
      {
        "action": "INITIALIZING",
        "objectID": "57653439117963557565",
        "objectName": "example",
        "objectType": "VOLUME"
      }
    ],
    "state": "NORMAL"
  },
  "status": {
    "success": true
  }
}
```

## 5.2 Version information

```
[GET] /v1/vroc/version
```

Returns a version of the installed VROC product, plugin version, Standard Storage Interface (SSI) information and trial information.

## Request parameters

*none*

## Response parameters (Object)

<b>passThroughSupport</b>	bool	Determines if this version of library supports pass through command.
<b>pluginVersion</b>	string	VROC plugin version.
<b>productVersion</b>	string	VROC product version.
<b>ssiInterfaceVersion</b>	string	Version of the interface used by SSI API library.
<b>ssiVersion</b>	string	SSI API library version.
<b>trial</b>	object	[Optional parameter] Trial information. Present if there is a VMD controller.
<b>{}.remainingDays</b>	integer	Indicates number of days that trial is available.

<code>{}.state</code>	string	Trial state. Possible values: NOT_STARTED, STARTED, EXPIRED, NOT_SUPPORTED.
<code>volSetSizeSupport</code>	bool	Determines if library supports setting volume cache size.
Available Errors		
<code>INVALID_PARAMETER</code>		Additional field's been attached with the request.
<code>VROC_OFFLINE</code>		VROC Plugin is not working.
<code>INTERNAL_ERROR</code>		An unexpected error occurred during request process.

## Request Example:

```
GET https://example.com/v1/vroc/version
```

```
{
  "data": {
    "passThroughSupport": false,
    "pluginVersion": "6.0.0.1072",
    "productVersion": "6.0.0.1072",
    "ssiInterfaceVersion": "1.0",
    "ssiVersion": "6.0.0.1072",
    "trial": {
      "remainingDays": 5,
      "state": "NOT_STARTED"
    },
    "volSetSizeSupport": false
  },
  "status": {
    "success": true
  }
}
```

## 5.3 Configuration information

```
[GET] /v1/vroc/config
```

Returns current configuration of VROC plugin. Each field in the configuration contains description and value. To modify specific fields in PUT request, use exactly the same names as in the response for this request.

## Request parameters

*none*

## Response parameters (Object)

<code>{configuration_field}</code>	object	Configuration field with unique name that contains information about the field.
<code>{}.description</code>	string	The description of the configuration field that explains its purpose.
<code>{}.max</code>	integer	<i>[Optional parameter]</i> Maximal value limitation. Present if the type is integer or float.

<code>{}.min</code>	integer	[Optional parameter] Minimal value limitation. Present if the type is integer or float.
<code>{}.type</code>	string	Value type that determines its' value. Possible values: integer, float, string, bool
<code>{}.value</code>	variant	Current value of the field.
Available Errors		
<b>INVALID_PARAMETER</b>		Additional field's been attached with the request.
<b>VROC_OFFLINE</b>		VROC Plugin is not working.
<b>INTERNAL_ERROR</b>		An unexpected error occurred during request process.

## Request Example:

```
GET https://example.com/v1/vroc/config
```

```
{
  "data": {
    "locate_timeout": {
      "description": "Defines timeout for LED Locate in seconds. When LED for port is on, it will flash for this amount of time and then turns off.",
      "max": 3600,
      "min": 1,
      "type": "integer",
      "value": 12
    }
  },
  "status": {
    "success": true
  }
}
```

## 5.4 List of RAIDs

```
[GET] /v1/vroc/raids[:raidId][?range={range}&[type={type}]
```

Returns list of supported RAID information structures.

## Request parameters

<b>raidId</b>	string	[Optional parameter] RAID identifier.
<b>range</b>	string	[Optional parameter] Scope of information details. Available values: info, level.
<b>type</b>	string	[Optional parameter] Name of the RAID level to filter information about level range. Valid only when 'range' is present and equals 'level'. Available values: RAID_0, RAID_1, RAID_5, RAID_10.

## Response parameters (Array)

<b>createFromExistingSupport</b>	bool	Determines if new volume can be created where the partitioned data on one of the member disks is preserved.
----------------------------------	------	---

<b>dedicatedSpareSupport</b>	bool	Spare disks can be created that will only rebuild a specific volume.
<b>emptyArraysSupport</b>	bool	Indicates that this controller supports empty arrays.
<b>globalSpareSupport</b>	bool	Spare disks can be created that are not associated with a specific array.
<b>maxDisksPerArray</b>	integer	The maximum number of disks allowed in an array.
<b>maxRaidDisksSupported</b>	integer	The maximum number of disks that are allowed to be used as array members.
<b>maxVolumesPerArray</b>	integer	The maximum number of volumes that can be created on an array.
<b>maxVolumesPerHba</b>	integer	The maximum number of volumes that can be created per controller.
<b>raidEnabled</b>	bool	True if the controller supports RAID, false if not.
<b>raidId</b>	string	RAID identifier.
<b>raidLevelInfo</b>	array	List of RAID level information objects.
<b>[].created</b>	bool	Indicates that there's at least one volume created with this RAID level.
<b>[].defaultStripSize</b>	string	<i>[Conditional parameter]</i> The default strip size to be used for this RAID level. If this field is present, <b>defaultStripSizes</b> is not. Condition: Present if RAID level is not RAID_5 and controller is not VMD. Possible values: SIZE_UNKNOWN, SIZE_4KB, SIZE_8KB, SIZE_16KB, SIZE_32KB, SIZE_64KB, SIZE_128KB, N/A.
<b>[].defaultStripSizes</b>	object	<i>[Conditional parameter]</i> Object of default strip sizes depending on a number of disks. If this field is present, <b>defaultStripSize</b> is not. Condition: Present if RAID level is RAID_5 and controller is VMD.
<b>[].{ }.{numberOfDisks}</b>	string	For each number of disks between minimum (e.g. 3) and maximum (e.g. 24), there is one default strip size. Possible values: SIZE_UNKNOWN, SIZE_4KB, SIZE_8KB, SIZE_16KB, SIZE_32KB, SIZE_64KB, SIZE_128KB.
<b>[].evenDiskCount</b>	bool	Indicates that this RAID level requires an even number of disks.
<b>[].maxDisks</b>	integer	The maximum number of disks supported by this RAID level.
<b>[].migrSupport</b>	array	Indicates all of the different RAID levels that a user can migrate to from this RAID level.
<b>[].[].raidLevel</b>	string	Name of the RAID level. Possible values: RAID_0, RAID_1, RAID_5, RAID_10.
<b>[].[].requireDisk</b>	bool	Indicates if RAID level requires a disk addition to perform a migration.
<b>[].minDisks</b>	integer	The minimum number of disks supported by this RAID level.
<b>[].oddDiskCount</b>	bool	Indicates that this RAID level requires an odd number of disks.

<code>[].raidLevel</code>	string	Name of the RAID level. Possible values: RAID_0, RAID_1, RAID_5, RAID_10.
<code>[].stripSizesSupported</code>	array	List of the strip sizes supported by this RAID level as strings. Can be empty. Possible values: SIZE_UNKNOWN, SIZE_4KB, SIZE_8KB, SIZE_16KB, SIZE_32KB, SIZE_64KB, SIZE_128KB.
Available Errors		
<code>INVALID_PARAMETER</code>		Invalid raidId or value of parameters or type is present when range is equal to info or unknown parameter is passed.
<code>VROC_OFFLINE</code>		VROC Plugin is not working.
<code>INTERNAL_ERROR</code>		An unexpected error occurred during request process.

## Request Example:

```
GET https://example.com/v1/vroc/raids/16722586989855889336?range=info
```

```
{
  "data": [
    {
      "createFromExistingSupport": true,
      "dedicatedSpareSupport": false,
      "emptyArraysSupport": false,
      "globalSpareSupport": true,
      "maxDisksPerArray": 24,
      "maxRaidDisksSupported": 256,
      "maxVolumesPerArray": 2,
      "maxVolumesPerHba": 24,
      "raidEnabled": true,
      "raidId": "16722586989855889336"
    }
  ],
  "status": {
    "success": true
  }
}
```

**GET** https://example.com/v1/vroc/raids?range=info

```
{
  "data": [
    {
      "createFromExistingSupport": true,
      "dedicatedSpareSupport": false,
      "emptyArraysSupport": false,
      "globalSpareSupport": true,
      "maxDisksPerArray": 8,
      "maxRaidDisksSupported": 8,
      "maxVolumesPerArray": 2,
      "maxVolumesPerHba": 8,
      "raidEnabled": true,
      "raidId": "16722586989855889336"
    },
    {
      ...
      "raidId": "16723543565243243873"
    },
    {
      ...
      "raidId": "16724500140359701002"
    }
  ],
  "status": {
    "success": true
  }
}
```

```
GET https://example.com/v1/vroc/raids/16724500140359701002?range=level&type=5
```

```
{
  "data": [
    {
      "raidId": "16724500140359701002",
      "raidLevelInfo": [
        {
          "created": false,
          "defaultStripSizes": {
            "3": "SIZE_64KB",
            "4": "SIZE_32KB",
            "5": "SIZE_128KB",
            "6": "SIZE_64KB",
            "7": "SIZE_128KB",
            "8": "SIZE_16KB",
            "9": "SIZE_128KB",
            "10": "SIZE_64KB",
            "11": "SIZE_128KB",
            "12": "SIZE_32KB",
            "13": "SIZE_128KB",
            "14": "SIZE_64KB",
            "15": "SIZE_128KB",
            "16": "SIZE_8KB",
            "17": "SIZE_128KB",
            "18": "SIZE_64KB",
            "19": "SIZE_128KB",
            "20": "SIZE_32KB",
            "21": "SIZE_128KB",
            "22": "SIZE_64KB",
            "23": "SIZE_128KB",
            "24": "SIZE_16KB"
          },
          "evenDiskCount": false,
          "maxDisks": 24,
          "migrSupport": [
            {
              "raidLevel": "RAID_5",
              "requiresDisk": false
            }
          ],
          "minDisks": 3,
          "oddDiskCount": false,
          "raidLevel": "RAID_5",
          "stripSizesSupported": [
            "SIZE_4KB",
            "SIZE_8KB",
            "SIZE_16KB",
            "SIZE_32KB",
            "SIZE_64KB",
            "SIZE_128KB"
          ]
        }
      ]
    }
  ],
  "status": {
    "success": true
  }
}
```

```
GET https://example.com/v1/vroc/raids/16724500140359701002?range=level
```

```
{
  "data": [
    {
      "raidId": "16724500140359701002",
      "raidLevelInfo": [
        {
          "created": false,
          "defaultStripSize": "SIZE_128KB",
          "evenDiskCount": false,
          "maxDisks": 24,
          "migrSupport": [
            {
              "raidLevel": "RAID_0",
              "requiresDisk": false
            },
            {
              "raidLevel": "RAID_5",
              "requiresDisk": true
            }
          ],
          "minDisks": 2,
          "oddDiskCount": false,
          "raidLevel": "RAID_0",
          "stripSizesSupported": [
            "SIZE_4KB",
            "SIZE_8KB",
            "SIZE_16KB",
            "SIZE_32KB",
            "SIZE_64KB",
            "SIZE_128KB"
          ]
        },
        {
          ...
          "raidLevel": "RAID_1",
          ...
        },
        {
          ...
          "raidLevel": "RAID_5",
          ...
        },
        {
          ...
          "raidLevel": "RAID_10",
          ...
        }
      ]
    }
  ],
  "status": {
    "success": true
  }
}
```

```
GET https://example.com/v1/vroc/raids?range=level
```

```
{
  "data": [
    {
      "raidId": "16722586989855889336",
      "raidLevelInfo": [
        { ... },
        { ... },
        { ... },
        { ... }
      ]
    },
    {
      "raidId": "16723543565243243873",
      "raidLevelInfo": [
        { ... },
        { ... },
        { ... },
        { ... }
      ]
    },
    {
      "raidId": "16724500140359701002",
      "raidLevelInfo": [
        { ... },
        { ... },
        { ... },
        { ... }
      ]
    }
  ],
  "status": {
    "success": true
  }
}
```

**GET** https://example.com/v1/vroc/raids/16722586989855889336

```
{
  "data": [
    {
      "createFromExistingSupport": true,
      "dedicatedSpareSupport": false,
      "emptyArraysSupport": false,
      "globalSpareSupport": true,
      "maxDisksPerArray": 8,
      "maxRaidDisksSupported": 8,
      "maxVolumesPerArray": 2,
      "maxVolumesPerHba": 8,
      "raidEnabled": true,
      "raidId": "16722586989855889336",
      "raidLevelInfo": [
        { ... },
        { ... },
        { ... },
        { ... }
      ]
    }
  ],
  "status": {
    "success": true
  }
}
```

```
GET https://example.com/v1/vroc/raids
```

```
{
  "data": [
    {
      "createFromExistingSupport": true,
      "dedicatedSpareSupport": false,
      "emptyArraysSupport": false,
      "globalSpareSupport": true,
      "maxDisksPerArray": 8,
      "maxRaidDisksSupported": 8,
      "maxVolumesPerArray": 2,
      "maxVolumesPerHba": 8,
      "raidEnabled": true,
      "raidId": "16722586989855889336",
      "raidLevelInfo": [
        { ... },
        { ... },
        { ... },
        { ... }
      ]
    },
    {
      ...
      "raidId": "16723543565243243873",
      ...
    },
    {
      ...
      "raidId": "16724500140359701002",
      ...
    }
  ],
  "status": {
    "success": true
  }
}
```

## 5.5 List of controllers associated with RAID

```
[GET] /v1/vroc/raids/:raidId/controllers
```

Returns list of all controllers associated with the RAID information structure.

Request parameters

Parameter	Type	Description
<b>raidId</b>	string	RAID identifier.

Response parameters (Array)

Parameter	Type	Description
<b>address</b>	string	[Optional parameter] Controller address. Only available on Linux.
<b>availableDevices</b>	integer	Number of available end devices in this controller.
<b>availableDisks</b>	integer	Number of available disks in this controller.
<b>deviceId</b>	string	Device ID as reported by PCI enumeration.
<b>disableESataSpanning</b>	bool	If true, arrays may not be created with mix of internal/external SATA disks.

<b>hardwareKey</b>	string	[ <i>Optional parameter</i> ] Installed hardware key on the platform. Only if type is VMD. Possible values: UNKNOWN, VROC_PASS_THRU, VROC_STANDARD, VROC_PREMIUM, VROC_INTEL_ONLY_SKU
<b>hardwareRevisionId</b>	integer	Revision ID as reported by PCI enumeration.
<b>id</b>	string	Controller identifier.
<b>mode</b>	string	Controller mode. Possible values: UNKNOWN, AHCI, RAID
<b>name</b>	string	Controller name.
<b>nvsramSupported</b>	bool	If true, controller supports NVSRAM.
<b>preboot</b>	object	Preboot options.
<b>{}.loaded</b>	bool	If true, OROM/EFI has been loaded.
<b>{}.managerVersion</b>	string	Version of the ROM or EFI driver.
<b>{}.twoTbDiskSupport</b>	bool	If true, OROM/EFI supports system disks greater than 2TB.
<b>{}.twoTbVolumeSupport</b>	true	If true, OROM/EFI supports system volumes greater than 2TB.
<b>raidId</b>	string	RAID id that this controller supports.
<b>readPatrol</b>	object	Read patrol information.
<b>{}.enabled</b>	bool	If true, controller has enabled read patrol.
<b>{}.supported</b>	bool	If true, controller supports read patrol.
<b>rohi</b>	object	Rebuild on Hot Insert options.
<b>{}.enabled</b>	bool	If true, controller has enabled Rebuild on Hot Insert.
<b>{}.supported</b>	bool	If true, controller supports Rebuild on Hot Insert.
<b>subClassCode</b>	integer	Subclass Code as reported by PCI enumeration.
<b>subSystemId</b>	string	Subsystem ID as reported by PCI enumeration.
<b>subVendorId</b>	string	Subvendor ID as reported by PCI enumeration.
<b>supportedFunctions</b>	object	List of supported functionalities.
<b>{}.addDiskToArray</b>	bool	If true, add disk to array action is available.
<b>{}.arrayCreate</b>	bool	If true, create array action is available.
<b>{}.markAsNormal</b>	bool	If true, mark as normal action is available.
<b>{}.markAsSpare</b>	bool	If true, mark as spare action is available.
<b>{}.portLocate</b>	bool	If true, port locate is available.
<b>{}.volumeCancelVerify</b>	bool	If true, cancel volume verify action is available.
<b>{}.volumeDelete</b>	bool	If true, delete volume action is available.
<b>{}.volumeModify</b>	bool	If true, modify volume action is available.
<b>{}.volumeRename</b>	bool	If true, rename volume action is available.
<b>{}.volumeSetCachePolicy</b>	bool	If true, set volume cache policy action is available.

<b>supportsTPV</b>	bool	[ <i>Optional parameter</i> ] If true, disks from third party vendors (not Intel) are supported. Only if type is VMD.
<b>type</b>	string	Controller type. Possible values: UNKNOWN, SATA, NVMe, VMD, SCU
<b>vendorId</b>	string	Vendor identifier as hex with 0x base.
<b>version</b>	string	Version of the driver component.
<b>xor</b>	object	XOR acceleration options.
<b>enabled</b>	bool	If true, controller has enabled H/W XOR acceleration.
<b>supported</b>	bool	If true, controller supports H/W XOR acceleration.
<b>Available Errors</b>		
<b>INVALID_PARAMETER</b>		Invalid raidId or unknown parameter is passed.
<b>VROC_OFFLINE</b>		VROC Plugin is not working.
<b>INTERNAL_ERROR</b>		An unexpected error occurred during request process.

## Request Example:

```
GET https://example.com/v1/vroc/raids/16722586989855889336/controllers
```

```
{
  "data": [
    {
      "availableDevices": 3,
      "availableDisks": 3,
      "deviceId": "0x201d",
      "disableESataSpanning": true,
      "hardwareKey": "VROC_PASS_THRU",
      "hardwareRevisionId": 4,
      "id": "14973794998208193328",
      "mode": "RAID",
      "name": "Intel(R) VROC (in pass-thru mode)",
      "nvsramSupported": false,
      "preboot": {
        "loaded": false,
        "managerVersion": "5.3.0.1041",
        "twoTbDiskSupport": true,
        "twoTbVolumeSupport": true
      },
      "raidId": "16724500140359701002",
      "readPatrol": {
        "enabled": false,
        "supported": false
      },
      "rohi": {
        "enabled": false,
        "supported": true
      },
      "subClassCode": 4,
      "subSystemId": "0x00",
      "subVendorId": "0x8086",
      "supportedFunctions": {
        "addDisksToArray": true,
        "arrayCreate": false,
        "markAsNormal": true,
        "markAsSpare": true,
        "portLocate": true,
        "volumeCancelVerify": true,
        "volumeDelete": true,
        "volumeModify": true,
        "volumeRename": true,
        "volumeSetCachePolicy": true
      },
      "supportsTPV": true,
      "type": "VMD",
      "vendorId": "0x8086",
      "version": "6.0.0.1112",
      "xor": {
        "enabled": false,
        "supported": false
      }
    }
  ],
  "status": {
    "success": true
  }
}
```

## 5.6 Controllers information

[GET] /v1/vroc/controllers		
Returns list of all controllers.		
Request parameters		
<i>none</i>		
Response parameters (Array)		
<b>address</b>	string	[Optional parameter] Controller address. Only available on Linux.
<b>availableDevices</b>	integer	Number of available end devices in this controller.
<b>availableDisks</b>	integer	Number of available disks in this controller.
<b>deviceId</b>	string	Device ID as reported by PCI enumeration.
<b>disableESataSpanning</b>	bool	If true, arrays may not be created with mix of internal/external SATA disks.
<b>hardwareKey</b>	string	[Optional parameter] Installed hardware key on the platform. Only if type is VMD. Possible values: UNKNOWN, VROC_PASS_THRU, VROC_STANDARD, VROC_PREMIUM, VROC_INTEL_ONLY_SKU
<b>hardwareRevisionId</b>	integer	Revision ID as reported by PCI enumeration.
<b>id</b>	string	Controller identifier.
<b>mode</b>	string	Controller mode. Possible values: UNKNOWN, AHCI, RAID
<b>name</b>	string	Controller name.
<b>nvsramSupported</b>	bool	If true, controller supports NVSRAM.
<b>preboot</b>	object	Preboot options.
<b>{}.loaded</b>	bool	If true, OROM/EFI has been loaded.
<b>{}.managerVersion</b>	string	Version of the ROM or EFI driver.
<b>{}.twoTbDiskSupport</b>	bool	If true, OROM/EFI supports system disks greater than 2TB.
<b>{}.twoTbVolumeSupport</b>	bool	If true, OROM/EFI supports system volumes greater than 2TB.
<b>raidId</b>	string	RAID id that this controller supports.
<b>readPatrol</b>	object	Read patrol information.
<b>{}.enabled</b>	bool	If true, controller has enabled read patrol.
<b>{}.supported</b>	bool	If true, controller supports read patrol.
<b>rohi</b>	object	Rebuild on Hot Insert options.
<b>{}.enabled</b>	bool	If true, controller has enabled Rebuild on Hot Insert.
<b>{}.supported</b>	bool	If true, controller supports Rebuild on Hot Insert.
<b>subClassCode</b>	integer	Subclass Code as reported by PCI enumeration.
<b>subSystemId</b>	string	Subsystem ID as reported by PCI enumeration.

<b>subVendorId</b>	string	Subvendor ID as reported by PCI enumeration.
<b>supportedFunctions</b>	object	List of supported functionalities.
<b>{}.addDiskToArray</b>	bool	If true, add disk to array action is available.
<b>{}.arrayCreate</b>	bool	If true, create array action is available.
<b>{}.markAsNormal</b>	bool	If true, mark as normal action is available.
<b>{}.markAsSpare</b>	bool	If true, mark as spare action is available.
<b>{}.portLocate</b>	bool	If true, port locate is available.
<b>{}.volumeCancelVerify</b>	bool	If true, cancel volume verify action is available.
<b>{}.volumeDelete</b>	bool	If true, delete volume action is available.
<b>{}.volumeModify</b>	bool	If true, modify volume action is available.
<b>{}.volumeRename</b>	bool	If true, rename volume action is available.
<b>{}.volumeSetCachePolicy</b>	bool	If true, set volume cache policy action is available.
<b>supportsTPV</b>	bool	[Optional parameter] If true, disks from third party vendors (not Intel) are supported. Only if type is VMD.
<b>type</b>	string	Controller type. Possible values: UNKNOWN, SATA, NVMe, VMD, SCU
<b>vendorId</b>	string	Vendor identifier as hex with 0x base.
<b>version</b>	string	Version of the driver component.
<b>xor</b>	object	XOR acceleration options.
<b>enabled</b>	bool	If true, controller has enabled H/W XOR acceleration.
<b>supported</b>	bool	If true, controller supports H/W XOR acceleration.
<b>Available Errors</b>		
<b>INVALID_PARAMETER</b>		Additional field's been attached with the request.
<b>VROC_OFFLINE</b>		VROC Plugin is not working.
<b>INTERNAL_ERROR</b>		An unexpected error occurred during request process.

## Request Example:

```
GET https://example.com/v1/vroc/controllers
```

```
{
  "data": [
    {
      "availableDevices": 6,
      "availableDisks": 6,
      "deviceId": "0x2826",
      "disableESataSpanning": true,
      "hardwareRevisionId": 3,
      "id": "1054866554645959075",
      "mode": "RAID",
      "name": "Intel(R) C600+/C220+ series chipset SATA RAID Controller",
      "nvsramSupported": false,
      "preboot": {
        "loaded": false,
        "managerVersion": "5.3.0.1041",
        "twoTbDiskSupport": true,
        "twoTbVolumeSupport": true
      },
      "raidId": "16723543565243243873",
      "readPatrol": {
        "enabled": false,
        "supported": true
      },
      "rohi": {
        "enabled": false,
        "supported": true
      },
      "subClassCode": 4,
      "subSystemId": "0x7270",
      "subVendorId": "0x8086",
      "supportedFunctions": {
        "addDisksToArray": true,
        "arrayCreate": false,
        "markAsNormal": true,
        "markAsSpare": true,
        "portLocate": true,
        "volumeCancelVerify": true,
        "volumeDelete": true,
        "volumeModify": true,
        "volumeRename": true,
        "volumeSetCachePolicy": true
      },
      "type": "SATA",
      "vendorId": "0x8086",
      "version": "6.0.0.1112",
      "xor": {
        "enabled": false,
        "supported": false
      }
    },
    {
      ...
      "id": "7648327465536555975",
      ...
    }
  ],
  "status": {
    "success": true
  }
}
```

## 5.7 End device information associated with controller

[GET] /v1/vroc/controllers/:controllerId/enddevices		
Returns list of all available end devices associated with specified controller.		
Request parameters		
<b>controllerId</b>	string	Controller identifier.
Response parameters (Array)		
<b>address</b>	string	Physical address of the end device. For SATA, it's SCSI address (D.D.D.D) and for VMD, it's BDF address (HHHH:HH.HH.H). D - decimal H - hexadecimal
<b>arrayId</b>	string	Associated array ID. Null if not in an array.
<b>controllerId</b>	string	Controller id to which device is connected.
<b>deviceType</b>	string	Device type. Possible values: UNKNOWN, DISK, NON_DISK_STORAGE, NON_STORAGE
<b>firmware</b>	string	Serial number of the device.
<b>freeSize</b>	string	[Optional parameter] Free size of the device in bytes. Only for DISK and NON_STORAGE devices.
<b>id</b>	string	End device identifier.
<b>isIntelNVMe</b>	bool	[Optional parameter] If true, device is NVMe with Intel vendor ID. Only for VMD type.
<b>managedHotplug</b>	bool	If true, device can be removed with managed hotplug functionality.
<b>model</b>	string	Serial number of the device.
<b>negotiatedLinkSpeed</b>	string	Link speed of the device. UNKNOWN if cannot be determined.
<b>raidId</b>	string	RAID identifier. Can be null if controller is in AHCI mode.
<b>sectorSize</b>	object	[Optional parameter] Sector size of the device. Only for DISK and NON_STORAGE devices.
<b>{}.logical</b>	string	Logical sector size in bytes.
<b>{}.physical</b>	string	Physical sector size in bytes.
<b>serialNumber</b>	string	Serial number of the device.
<b>state</b>	string	Disk current state. Possible values: UNKNOWN, OFFLINE, MISSING, FAILED, SMART_EVENT_TRIGGERED, CONFIG_IS_UPREV, NORMAL, LOCKED, MANUAL_OFFLINE, INCOMPATIBLE, UNSUPPORTED
<b>systemDisk</b>	bool	[Optional parameter] If true, this device holds OS. Only for DISK and NON_STORAGE devices.
<b>totalSize</b>	string	[Optional parameter] Size of the device in bytes. Only for DISK and NON_STORAGE devices.

<b>type</b>	string	Device type. Possible values: UNKNOWN, SATA, NVME, VMD
<b>usage</b>	string	Disk current usage. Possible values: UNKNOWN, ARRAY_MEMBER, PASS_THROUGH, OFFLINE_ARRAY, SPARE, ARRAY_MEMBER_READONLY_MOUNT, PASS_THRU_READONLY_MOUNT, JOURNALING_DRIVE
<b>vendorId</b>	string	[ <i>Optional parameter</i> ] Vendor ID of the device. Only for VMD type.
<b>vmdDomain</b>	integer	[ <i>Optional parameter</i> ] VMD domain of the device. Only for VMD type.
<b>writeCachePolicy</b>	string	[ <i>Optional parameter</i> ] Write Cache Policy status. Only for DISK and NON_STORAGE devices. Possible values: UNKNOWN, ON, OFF, NOT_SUPPORTED
<b>x8Disk</b>	string	[ <i>Optional parameter</i> ] Determines if device is first or second of dual disk NVMe drive. Only for DISK and NON_STORAGE devices which are NVMe x8 dual disk drive.
<b>Available Errors</b>		
<b>INVALID_PARAMETER</b>	Invalid controllerId or unknown parameter is passed.	
<b>OBJECT_NOT_FOUND</b>	ControllerId cannot be found.	
<b>VROC_OFFLINE</b>	VROC Plugin is not working.	
<b>INTERNAL_ERROR</b>	An unexpected error occurred during request process.	

## Request Example:

```
GET https://example.com/v1/vroc/controllers/546984345282310681/enddevices
```

```
{
  "data": [
    { ... },
    {
      "address": "5-0-3-0",
      "arrayId": "1695433104163244970",
      "controllerId": "546984345282310681",
      "deviceType": "DISK",
      ...
      "id": "1183373271471488863",
      "isIntelNVMe": true,
      "managedHotplug": true,
      "model": "INTEL SSDPEDMD400G4",
      ...
      "serialNumber": "CVFT534300K0400BGN",
      ...
    },
    { ... },
  ],
  "status": {
    "success": true
  }
}
```

## 5.8 List of arrays associated with controller

```
[GET] /v1/vroc/controllers/:controllerId/arrays
```

Returns list of all available arrays associated with specified controller.

## Request parameters

parameter	type	description
<b>controllerId</b>	string	Controller identifier.

## Response parameters (Array)

parameter	type	description
<b>controllerIds</b>	array	Array of strings which represent controller IDs associated with the array.
<b>freeSize</b>	string	Free size of the array in bytes.
<b>id</b>	string	Array identifier.
<b>name</b>	string	Name of the array.
<b>numDisks</b>	integer	Number of disks inside the array.
<b>numVolumes</b>	integer	Number of volumes inside the array.
<b>state</b>	string	Array current state. Possible values: UNKNOWN, STATE_BUSY, NORMAL
<b>totalSize</b>	string	Total size of the array in bytes.
<b>writeCachePolicy</b>	string	Write Cache Policy status. Possible values: UNKNOWN, ON, OFF, NOT_SUPPORTED

Available Errors	
<b>INVALID_PARAMETER</b>	Invalid controllerId or unknown parameter is passed.
<b>OBJECT_NOT_FOUND</b>	ControllerId cannot be found.
<b>VROC_OFFLINE</b>	VROC Plugin is not working.
<b>INTERNAL_ERROR</b>	An unexpected error occurred during request process.

## Request Example:

<b>GET</b> https://example.com/v1/vroc/controllers/546984345282310681/arrays
<pre>{   "data": [     {       "controllerIds": [         "546984345282310681"       ],       "freeSize": "607116066816",       "id": "1695433104163244970",       "name": "NVMe*_Array_0044",       ...     }   ],   "status": {     "success": true   } }</pre>

## 5.9 Ports information associated with controller

<b>[GET] /v1/vroc/controllers/:controllerId/ports</b>		
Returns list of all available ports associated with specified controller.		
Request parameters		
<b>controllerId</b>	string	Controller identifier.
Response parameters (Array)		
<b>address</b>	string	Port physical address.
<b>connectedPortId</b>	string	Id of port that is connected to.
<b>id</b>	string	Port identifier.
<b>localDeviceId</b>	string	Device Id connected to the port.
<b>localDeviceType</b>	string	Type of device. Possible values: UNKNOWN, CONTROLLER, END_DEVICE
<b>numPhys</b>	integer	The width of the port defined by the number of Phys: x1, x2 or x4.
Available Errors		
<b>INVALID_PARAMETER</b>	Invalid controllerId or unknown parameter is passed.	

VROC_OFFLINE	VROC Plugin is not working.
INTERNAL_ERROR	An unexpected error occurred during request process.

## Request Example:

```
GET https://example.com/v1/vroc/controllers/14973794998208193328/ports
```

```
{
  "data": [
    { ... },
    {
      "address": "3-0-0-0",
      "connectedPortId": "13893375693132011551",
      "id": "13898158568710834882",
      "localDeviceId": "14973794998208193328",
      "localDeviceType": "CONTROLLER",
      "numPhys": 1
    },
    { ... },
  ],
  "status": {
    "success": true
  }
}
```

## 5.10 List of volumes associated with controller

```
[GET] /v1/vroc/controllers/:controllerId/volumes
```

Returns list of all available volumes associated with specified controller.

## Request parameters

<b>controllerId</b>	string	Controller identifier.
---------------------	--------	------------------------

## Response parameters (Array)

<b>arrayId</b>	string	Array identifier associated with the volume.
<b>arrayOrdinal</b>	integer	Position of the volume in the array.
<b>cachePolicy</b>	string	Current Cache Policy of the volume. This is not supported for Linux. Possible values: UNKNOWN, OFF, WRITE_THROUGH, WRITE_BACK, NOT_SUPPORTED
<b>id</b>	string	Volume identifier.
<b>initialized</b>	bool	If true, RAID mechanisms are fully initialized on the volume.
<b>journalingDriveId</b>	string	[Optional parameter] Identifier of the journaling drive associated to the volume. Only for RAID_5 and RWH Policy set to RWH_JOURNALING_DRIVE under Windows. This is not supported for Linux.
<b>migrProgress</b>	integer	Percentage progress for a running operation like migration, expansion or verify. Only valid if migration is true.
<b>migration</b>	bool	If true, the volume is during a operation and migrProgress shows meaningful data.

<b>name</b>	string	Volume name. Contains no more than 16 ASCII characters from range 32 to 126 (inclusive) without backslash (92). Spaces (20) are available but cannot be at the beginning. For Linux, also slash (47) is unavailable.
<b>numDisks</b>	integer	Number of disks used by the volume.
<b>raidLevel</b>	string	RAID level of the volume. Possible values: RAID_UNKNOWN, RAID_INVALID, RAID_0, RAID_1, RAID_5, RAID_10
<b>rwhPolicy</b>	string	[ <i>Optional parameter</i> ] Current RAID Write Hole policy. Only for RAID_5. Possible values: RWH_OFF, RWH_DISTRIBUTED, RWH_JOURNALING_DRIVE
<b>sectorSize</b>	object	Sector size of the volume.
<b>{}.logical</b>	string	Logical sector size in bytes.
<b>{}.physical</b>	string	Physical sector size in bytes.
<b>state</b>	string	Current volume state. Possible values: UNKNOWN, NORMAL, DEGRADED, FAILED, INITIALIZING, REBUILDING, VERIFYING, VERIFYING_AND_FIXING, GENERAL_MIGRATION, LOCKED, NON_REDUNDANT_VOLUME_FAILED_DISK
<b>stripSize</b>	string	Strip size of the volume. Possible values: SIZE_UNKNOWN, SIZE_4KB, SIZE_8KB, SIZE_16KB, SIZE_32KB, SIZE_64KB, SIZE_128KB
<b>systemVolume</b>	bool	If true, this volume holds OS.
<b>totalSize</b>	string	Total size of the volume in bytes.
<b>verify</b>	object	Last verify or verify and fix operation status.
<b>{}.badBlocks</b>	integer	Number of bad blocks found.
<b>{}.errors</b>	integer	Number of parity errors found.
<b>Available Errors</b>		
<b>INVALID_PARAMETER</b>		Invalid controllerId or unknown parameter is passed.
<b>OBJECT_NOT_FOUND</b>		ControllerId cannot be found.
<b>VROC_OFFLINE</b>		VROC Plugin is not working.
<b>INTERNAL_ERROR</b>		An unexpected error occurred during request process.

## Request Example:

```
GET https://example.com/v1/vroc/controllers/7826705849206250768/volumes
```

```
{
  "data": [
    {
      "arrayId": "14653709924164455516",
      "arrayOrdinal": 0,
      "cachePolicy": "NOT_SUPPORTED",
      "id": "3251927784729082717",
      "initialized": false,
      "migrProgress": 5,
      "migration": true,
      "name": "Volume_001",
      "numDisks": 4,
      "raidLevel": "RAID_5",
      "rwhPolicy": "RWH_OFF",
      "sectorSize": {
        "logical": "512",
        "physical": "512"
      },
      "state": "INITIALIZING",
      "stripSize": "SIZE_32KB",
      "systemVolume": false,
      "totalSize": "214746267648",
      "verify": {
        "badBlocks": 0,
        "errors": 1975192
      }
    }
  ],
  "status": {
    "success": true
  }
}
```

## 5.11 Ports information

```
[GET] /v1/vroc/ports
```

Returns list of all available ports.

## Request parameters

*none*

## Response parameters (Array)

Parameter	Type	Description
<b>address</b>	string	Port physical address.
<b>connectedPortId</b>	string	Id of port that is connected to.
<b>id</b>	string	Port identifier.
<b>localDeviceId</b>	string	Device Id connected to the port.
<b>localDeviceType</b>	string	Type of device. Possible values: UNKNOWN, CONTROLLER, END_DEVICE
<b>numPhys</b>	integer	The width of the port defined by the number of Phys: x1, x2 or x4.

Available Errors	
<b>INVALID_PARAMETER</b>	Additional field's been attached with the request.
<b>VROC_OFFLINE</b>	VROC Plugin is not working.
<b>INTERNAL_ERROR</b>	An unexpected error occurred during request process.

## Request Example:

```
GET https://example.com/v1/vroc/ports
```

```
{
  "data": [
    {
      "address": "1-0-0-0",
      "connectedPortId": "2309222417023443211",
      "id": "9186161837211096428",
      "localDeviceId": "1054866554645959075",
      "localDeviceType": "CONTROLLER",
      "numPhys": 1
    },
    {
      "address": "1-1-0-0",
      "connectedPortId": "2310178992139494434",
      "id": "9185205262094504517",
      "localDeviceId": "1054866554645959075",
      "localDeviceType": "CONTROLLER",
      "numPhys": 1
    },
    {
      "address": "1-0-0-0",
      "connectedPortId": "9186161837211096428",
      "id": "2309222417023443211",
      "localDeviceId": "6908014343806162082",
      "localDeviceType": "END_DEVICE",
      "numPhys": 1
    },
    {
      "address": "1-1-0-0",
      "connectedPortId": "9185205262094504517",
      "id": "2310178992139494434",
      "localDeviceId": "6029574419765821177",
      "localDeviceType": "END_DEVICE",
      "numPhys": 1
    }
  ],
  "status": {
    "success": true
  }
}
```

## 5.12 End device information

[GET] /v1/vroc/enddevices		
Returns list of all available end devices.		
Request parameters		
<i>none</i>		
Response parameters (Array)		
<b>address</b>	string	Physical address of the end device. For SATA, it's SCSI address (D.D.D.D) and for VMD, it's BDF address (HHHH:HH.HH.H). D - decimal H - hexadecimal
<b>arrayId</b>	string	Associated array ID. Null if not in an array.
<b>controllerId</b>	string	Controller id to which device is connected.
<b>deviceType</b>	string	Device type. Possible values: UNKNOWN, DISK, NON_DISK_STORAGE, NON_STORAGE
<b>firmware</b>	string	Serial number of the device.
<b>freeSize</b>	string	[Optional parameter] Free size of the device in bytes. Only for DISK and NON_STORAGE devices.
<b>id</b>	string	End device identifier.
<b>isIntelNVMe</b>	bool	[Optional parameter] If true, device is NVMe with Intel vendor ID. Only for VMD type.
<b>managedHotplug</b>	bool	If true, device can be removed with managed hotplug functionality.
<b>model</b>	string	Serial number of the device.
<b>negotiatedLinkSpeed</b>	string	Link speed of the device. UNKNOWN if cannot be determined.
<b>raidId</b>	string	RAID identifier. Can be null if controller is in AHCI mode.
<b>sectorSize</b>	object	[Optional parameter] Sector size of the device. Only for DISK and NON_STORAGE devices.
<b>{}.logical</b>	string	Logical sector size in bytes.
<b>{}.physical</b>	string	Physical sector size in bytes.
<b>serialNumber</b>	string	Serial number of the device.
<b>state</b>	string	Disk current state. Possible values: UNKNOWN, OFFLINE, MISSING, FAILED, SMART_EVENT_TRIGGERED, CONFIG_IS_UPREV, NORMAL, LOCKED, MANUAL_OFFLINE, INCOMPATIBLE, UNSUPPORTED
<b>systemDisk</b>	bool	[Optional parameter] If true, this device holds OS. Only for DISK and NON_STORAGE devices.
<b>totalSize</b>	string	[Optional parameter] Size of the device in bytes. Only for DISK and NON_STORAGE devices.

<b>type</b>	string	Device type. Possible values: UNKNOWN, SATA, NVME, VMD
<b>usage</b>	string	Disk current usage. Possible values: UNKNOWN, ARRAY_MEMBER, PASS_THROUGH, OFFLINE_ARRAY, SPARE, ARRAY_MEMBER_READONLY_MOUNT, PASS_THRU_READONLY_MOUNT, JOURNALING_DRIVE
<b>vendorId</b>	string	[ <i>Optional parameter</i> ] Vendor ID of the device. Only for VMD type.
<b>vmdDomain</b>	integer	[ <i>Optional parameter</i> ] VMD domain of the device. Only for VMD type.
<b>writeCachePolicy</b>	string	[ <i>Optional parameter</i> ] Write Cache Policy status. Only for DISK and NON_STORAGE devices. Possible values: UNKNOWN, ON, OFF, NOT_SUPPORTED
<b>x8Disk</b>	string	[ <i>Optional parameter</i> ] Determines if device is first or second of dual disk NVMe drive. Only for DISK and NON_STORAGE devices which are NVMe x8 dual disk drive.
<b>Available Errors</b>		
<b>INVALID_PARAMETER</b>		Additional field's been attached with the request.
<b>VROC_OFFLINE</b>		VROC Plugin is not working.
<b>INTERNAL_ERROR</b>		An unexpected error occurred during request process.

## Request Example:

```
GET https://example.com/v1/vroc/enddevices
```

```
{
  "data": [
    {
      "address": "2-0-0-0",
      "arrayId": null,
      "controllerId": "546984345282310681",
      "deviceType": "DISK",
      "firmware": "8DV101H0",
      "freeSize": "800166076416",
      "id": "6122016497971580907",
      "isIntelNVMe": true,
      "managedHotplug": true,
      "model": "INTEL SSDPECME016T4",
      "negotiatedLinkSpeed": "UNKNOWN",
      "raidId": "5247921164796271273",
      "sectorSize": {
        "logical": "512",
        "physical": "512"
      },
      "serialNumber": "CVF85504008H1P6BGN-1",
      "state": "NORMAL",
      "systemDisk": false,
      "totalSize": "800166076416",
      "type": "VMD",
      "usage": "PASS_THROUGH",
      "vendorId": "0x8086",
      "vmdDomain": 2,
      "writeCachePolicy": "ON",
      "x8Disk": "X8_A"
    }
  ],
  "status": {
    "success": true
  }
}
```

## 5.13 List of all arrays

```
[GET] /v1/vroc/arrays
```

Returns list of all available arrays.

## Request parameters

*none*

## Response parameters (Array)

<b>controllerIds</b>	array	Array of strings which represent controller IDs associated with the array.
<b>freeSize</b>	string	Free size of the array in bytes.
<b>id</b>	string	Array identifier.
<b>name</b>	string	Name of the array.
<b>numDisks</b>	integer	Number of disks inside the array.
<b>numVolumes</b>	integer	Number of volumes inside the array.

<b>state</b>	string	Array current state. Possible values: UNKNOWN, STATE_BUSY, NORMAL
<b>totalSize</b>	string	Total size of the array in bytes.
<b>writeCachePolicy</b>	string	Write Cache Policy status. Possible values: UNKNOWN, ON, OFF, NOT_SUPPORTED
<b>Available Errors</b>		
<b>INVALID_PARAMETER</b>		Additional field's been attached with the request.
<b>VROC_OFFLINE</b>		VROC Plugin is not working.
<b>INTERNAL_ERROR</b>		An unexpected error occurred during request process.

## Request Example:

<b>GET</b> https://example.com/v1/vroc/arrays
<pre>{   "data": [     {       "controllerIds": [         "546984345282310681"       ],       "freeSize": "607116066816",       "id": "1695433104163244970",       "name": "NVMe*_Array_0044",       "numDisks": 3,       "numVolumes": 1,       "state": "STATE_BUSY",       "totalSize": "768180486144",       "writeCachePolicy": "ON"     }   ],   "status": {     "success": true   } }</pre>

## 5.14 End device information associated with array

<b>[GET] /v1/vroc/arrays/:arrayId/enddevices</b>		
Returns list of all available end devices associated with specified array.		
<b>Request parameters</b>		
<b>arrayId</b>	string	Array identifier.
<b>Response parameters (Array)</b>		
<b>address</b>	string	Physical address of the end device. For SATA, it's SCSI address (D.D.D.D) and for VMD, it's BDF address (HHH:HH.HH.H). D - decimal H - hexadecimal

<b>arrayId</b>	string	Associated array ID. Null if not in an array.
<b>controllerId</b>	string	Controller id to which device is connected.
<b>deviceType</b>	string	Device type. Possible values: UNKNOWN, DISK, NON_DISK_STORAGE, NON_STORAGE
<b>firmware</b>	string	Serial number of the device.
<b>freeSize</b>	string	[Optional parameter] Free size of the device in bytes. Only for DISK and NON_STORAGE devices.
<b>id</b>	string	End device identifier.
<b>isIntelNVMe</b>	bool	[Optional parameter] If true, device is NVMe with Intel vendor ID. Only for VMD type.
<b>managedHotplug</b>	bool	If true, device can be removed with managed hotplug functionality.
<b>model</b>	string	Serial number of the device.
<b>negotiatedLinkSpeed</b>	string	Link speed of the device. UNKNOWN if cannot be determined.
<b>raidId</b>	string	RAID identifier. Can be null if controller is in AHCI mode.
<b>sectorSize</b>	object	[Optional parameter] Sector size of the device. Only for DISK and NON_STORAGE devices.
<b>{}.logical</b>	string	Logical sector size in bytes.
<b>{}.physical</b>	string	Physical sector size in bytes.
<b>serialNumber</b>	string	Serial number of the device.
<b>state</b>	string	Disk current state. Possible values: UNKNOWN, OFFLINE, MISSING, FAILED, SMART_EVENT_TRIGGERED, CONFIG_IS_UPREV, NORMAL, LOCKED, MANUAL_OFFLINE, INCOMPATIBLE, UNSUPPORTED
<b>systemDisk</b>	bool	[Optional parameter] If true, this device holds OS. Only for DISK and NON_STORAGE devices.
<b>totalSize</b>	string	[Optional parameter] Size of the device in bytes. Only for DISK and NON_STORAGE devices.
<b>type</b>	string	Device type. Possible values: UNKNOWN, SATA, NVME, VMD
<b>usage</b>	string	Disk current usage. Possible values: UNKNOWN, ARRAY_MEMBER, PASS_THROUGH, OFFLINE_ARRAY, SPARE, ARRAY_MEMBER_READONLY_MOUNT, PASS_THRU_READONLY_MOUNT, JOURNALING_DRIVE
<b>vendorId</b>	string	[Optional parameter] Vendor ID of the device. Only for VMD type.
<b>vmdDomain</b>	integer	[Optional parameter] VMD domain of the device. Only for VMD type.

<b>writeCachePolicy</b>	string	[Optional parameter] Write Cache Policy status. Only for DISK and NON_STORAGE devices. Possible values: UNKNOWN, ON, OFF, NOT_SUPPORTED
<b>x8Disk</b>	string	[Optional parameter] Determines if device is first or second of dual disk NVMe drive. Only for DISK and NON_STORAGE devices which are NVMe x8 dual disk drive.
<b>Available Errors</b>		
<b>INVALID_PARAMETER</b>		Invalid arrayId or unknown parameter is passed.
<b>OBJECT_NOT_FOUND</b>		ArrayId cannot be found.
<b>VROC_OFFLINE</b>		VROC Plugin is not working.
<b>INTERNAL_ERROR</b>		An unexpected error occurred during request process.

## Request Example:

```
GET https://example.com/v1/vroc/arrays/1695433104163244970/enddevices
```

```
{
  "data": [
    { ... },
    {
      "address": "5-0-1-0",
      "arrayId": "1695433104163244970",
      ...
      "id": "1958265928247903799",
      ...
      "model": "INTEL SSDPEDMD400G4",
      ...
      "serialNumber": "CVFT5155004F400BGN",
      ...
    },
    { ... },
  ],
  "status": {
    "success": true
  }
}
```

## 5.15 List of volumes associated with array

[GET] /v1/vroc/arrays/:arrayId/volumes		
Returns list of all available volumes associated with specified array.		
Request parameters		
<b>arrayId</b>	string	Array identifier.
Response parameters (Array)		
<b>arrayId</b>	string	Array identifier associated with the volume.
<b>arrayOrdinal</b>	integer	Position of the volume in the array.
<b>cachePolicy</b>	string	Current Cache Policy of the volume. This is not supported for Linux. Possible values: UNKNOWN, OFF, WRITE_THROUGH, WRITE_BACK, NOT_SUPPORTED
<b>id</b>	string	Volume identifier.
<b>initialized</b>	bool	If true, RAID mechanisms are fully initialized on the volume.
<b>journalingDriveId</b>	string	[Optional parameter] Identifier of the journaling drive associated to the volume. Only for RAID_5 and RWH Policy set to RWH_JOURNALING_DRIVE under Windows. This is not supported for Linux.
<b>migrProgress</b>	integer	Percentage progress for a running operation like migration, expansion or verify. Only valid if migration is true.
<b>migration</b>	bool	If true, the volume is during a operation and migrProgress shows meaningful data.
<b>name</b>	string	Volume name. Contains no more than 16 ASCII characters from range 32 to 126 (inclusive) without backslash (92). Spaces (20) are available but cannot be at the beginning. For Linux, also slash (47) is unavailable.
<b>numDisks</b>	integer	Number of disks used by the volume.
<b>raidLevel</b>	string	RAID level of the volume. Possible values: RAID_UNKNOWN, RAID_INVALID, RAID_0, RAID_1, RAID_5, RAID_10
<b>rwhPolicy</b>	string	[Optional parameter] Current RAID Write Hole policy. Only for RAID_5. Possible values: RWH_OFF, RWH_DISTRIBUTED, RWH_JOURNALING_DRIVE
<b>sectorSize</b>	object	Sector size of the volume.
<b>{}.logical</b>	string	Logical sector size in bytes.
<b>{}.physical</b>	string	Physical sector size in bytes.
<b>state</b>	string	Current volume state. Possible values: UNKNOWN, NORMAL, DEGRADED, FAILED, INITIALIZING, REBUILDING, VERIFYING, VERIFYING_AND_FIXING, GENERAL_MIGRATION, LOCKED, NON_REDUNDANT_VOLUME_FAILED_DISK

<b>stripSize</b>	string	Strip size of the volume. Possible values: SIZE_UNKNOWN, SIZE_4KB, SIZE_8KB, SIZE_16KB, SIZE_32KB, SIZE_64KB, SIZE_128KB
<b>systemVolume</b>	bool	If true, this volume holds OS.
<b>totalSize</b>	string	Total size of the volume in bytes.
<b>verify</b>	object	Last verify or verify and fix operation status.
<b>{}.badBlocks</b>	integer	Number of bad blocks found.
<b>{}.errors</b>	integer	Number of parity errors found.
<b>Available Errors</b>		
<b>INVALID_PARAMETER</b>		Invalid arrayId or unknown parameter is passed.
<b>OBJECT_NOT_FOUND</b>		ArrayId cannot be found.
<b>VROC_OFFLINE</b>		VROC Plugin is not working.
<b>INTERNAL_ERROR</b>		An unexpected error occurred during request process.

## Request Example:

```
GET https://example.com/v1/vroc/arrays/15568883836126137805/volumes
```

```
{
  "data": [
    {
      "arrayId": "1695433104163244970",
      "arrayOrdinal": 0,
      "cachePolicy": "OFF",
      "id": "10626977953712277614",
      "initialized": true,
      "journalingDriveId": "11651016949732315554",
      "migrProgress": 0,
      "migration": false,
      "name": "Volume_001",
      "numDisks": 3,
      "raidLevel": "RAID_5",
      "rwhPolicy": "RWH_JOURNALING_DRIVE",
      "sectorSize": {
        "logical": "512",
        "physical": "512"
      },
      "state": "NORMAL",
      "stripSize": "SIZE_64KB",
      "systemVolume": false,
      "totalSize": "107374182400",
      "verify": {
        "badBlocks": 0,
        "errors": 0
      }
    }
  ],
  "status": {
    "success": true
  }
}
```

## 5.16 List of all volumes

[GET] /v1/vroc/volumes		
Returns list of all available volumes.		
Request parameters		
<i>none</i>		
Response parameters (Array)		
<b>arrayId</b>	string	Array identifier associated with the volume.
<b>arrayOrdinal</b>	integer	Position of the volume in the array.
<b>cachePolicy</b>	string	Current Cache Policy of the volume. This is not supported for Linux. Possible values: UNKNOWN, OFF, WRITE_THROUGH, WRITE_BACK, NOT_SUPPORTED
<b>id</b>	string	Volume identifier.
<b>initialized</b>	bool	If true, RAID mechanisms are fully initialized on the volume.
<b>journalingDriveId</b>	string	[Optional parameter] Identifier of the journaling drive associated to the volume. Only for RAID_5 and RWH Policy set to RWH_JOURNALING_DRIVE under Windows. This is not supported for Linux.
<b>migrProgress</b>	integer	Percentage progress for a running operation like migration, expansion or verify. Only valid if migration is true.
<b>migration</b>	bool	If true, the volume is during a operation and migrProgress shows meaningful data.
<b>name</b>	string	Volume name. Contains no more than 16 ASCII characters from range 32 to 126 (inclusive) without backslash (92). Spaces (20) are available but cannot be at the beginning. For Linux, also slash (47) is unavailable.
<b>numDisks</b>	integer	Number of disks used by the volume.
<b>raidLevel</b>	string	RAID level of the volume. Possible values: RAID_UNKNOWN, RAID_INVALID, RAID_0, RAID_1, RAID_5, RAID_10
<b>rwhPolicy</b>	string	[Optional parameter] Current RAID Write Hole policy. Only for RAID_5. Possible values: RWH_OFF, RWH_DISTRIBUTED, RWH_JOURNALING_DRIVE
<b>sectorSize</b>	object	Sector size of the volume.
<b>{}.logical</b>	string	Logical sector size in bytes.
<b>{}.physical</b>	string	Physical sector size in bytes.
<b>state</b>	string	Current volume state. Possible values: UNKNOWN, NORMAL, DEGRADED, FAILED, INITIALIZING, REBUILDING, VERIFYING, VERIFYING_AND_FIXING, GENERAL_MIGRATION, LOCKED, NON_REDUNDANT_VOLUME_FAILED_DISK

<b>stripSize</b>	string	Strip size of the volume. Possible values: SIZE_UNKNOWN, SIZE_4KB, SIZE_8KB, SIZE_16KB, SIZE_32KB, SIZE_64KB, SIZE_128KB
<b>systemVolume</b>	bool	If true, this volume holds OS.
<b>totalSize</b>	string	Total size of the volume in bytes.
<b>verify</b>	object	Last verify or verify and fix operation status.
<b>{}.badBlocks</b>	integer	Number of bad blocks found.
<b>{}.errors</b>	integer	Number of parity errors found.
<b>Available Errors</b>		
<b>INVALID_PARAMETER</b>		Additional field's been attached with the request.
<b>VROC_OFFLINE</b>		VROC Plugin is not working.
<b>INTERNAL_ERROR</b>		An unexpected error occurred during request process.

## Request Example:

```
GET https://example.com/v1/vroc/volumes
```

```
{
  "data": [
    {
      "arrayId": "1695433104163244970",
      "arrayOrdinal": 0,
      "cachePolicy": "OFF",
      "id": "10626977953712277614",
      "initialized": true,
      "journalingDriveId": "11651016949732315554",
      "migrProgress": 0,
      "migration": false,
      "name": "Volume_001",
      "numDisks": 3,
      "raidLevel": "RAID_5",
      "rwhPolicy": "RWH_JOURNALING_DRIVE",
      "sectorSize": {
        "logical": "512",
        "physical": "512"
      },
      "state": "NORMAL",
      "stripSize": "SIZE_64KB",
      "systemVolume": false,
      "totalSize": "107374182400",
      "verify": {
        "badBlocks": 0,
        "errors": 0
      }
    }
  ],
  "status": {
    "success": true
  }
}
```

## 5.17 Events information

[GET] /v1/vroc/events?time={time}		
Returns list of events that happened in specified time.		
Request parameters		
<b>time</b>	string	Time of events' occurrence in seconds.
Response parameters (Array)		
<b>creation_time</b>	string	Date of event's occurrence.
<b>description</b>	string	Readable description of the event.
<b>event_source</b>	string	Source of the event. Possible values: DISK, VOLUME
<b>event_source_name</b>	string	Name of the object that sends the event.
<b>event_type</b>	string	Type of the event. Possible values for DISK: DISK_UNKNOWN, DISK_SMART, DISK_FAILED, DISK_UNLOCKED, DISK_ADDED, DISK_REMOVED, DISK_MISSING Possible values for VOLUME: VOLUME_UNKNOWN, VOLUME_ADDED, VOLUME_REBUILD_STARTED, VOLUME_REBUILD_COMPLETE, VOLUME_VERIFY_STARTED, VOLUME_VERIFY_STOP, VOLUME_VERIFY_AND_FIX_STARTED, VOLUME_VERIFY_AND_FIX_STOP, VOLUME_MIGRATION_STARTED, VOLUME_MIGRATION_COMPLETE, VOLUME_INITIALIZE_STARTED, VOLUME_INITIALIZE_COMPLETE, VOLUME_DEGRADED, VOLUME_FAILED, VOLUME_UNLOCKED, VOLUME_LOCKED, VOLUME_DELETED, VOLUME_SYNCING_STARTED, VOLUME_SYNCING_COMPLETED, VOLUME_RECOVERY_DISK_DATA_INVALID, VOLUME_MASTER_DISK_DATA_INVALID, VOLUME_MASTER_DISK_FAILED, VOLUME_RECOVERY_DISK_FAILED, VOLUME_REVERSE_SYNC_STARTED, VOLUME_REVERSE_SYNC_COMPLETE, VOLUME_NORMAL
<b>severity</b>	string	Category of the event. Possible values: NONE, INFO, WARNING, ERROR, UNDEFINED
Available Errors		
<b>INVALID_PARAMETER</b>		Invalid value of time or time parameter is missing or unknown parameter is passed.
<b>VROC_OFFLINE</b>		VROC Plugin is not working.
<b>INTERNAL_ERROR</b>		An unexpected error occurred during request process.

## Request Example:

```
GET https://example.com/v1/vroc/events?time=3600
```

```
{
  "data": [
    {
      "creation_time": "03/10/98 01:02:41 AM",
      "description": "Volume_001 initialization was completed",
      "event_source": "VOLUME",
      "event_source_name": "Volume_001",
      "event_type": "VOLUME_INITIALIZE_COMPLETE",
      "severity": "INFO"
    }
  ],
  "status": {
    "success": true
  }
}
```

## 6. Actions

### 6.1 Edit configuration

```
[PUT] /v1/vroc/config
```

Edit configuration fields by providing their new values. To get the list of available fields names and their types, use GET request for config.

#### Request parameters (Object)

<b>{configuration_field}</b>	variant	Configuration field with unique name and a value to be set. There can be multiple fields to configure at once.
------------------------------	---------	--

#### Response parameters (Object)

*none*

#### Available Errors

<b>INVALID_PARAMETER</b>	There's no configuration fields or field does not exist or field's value does not meet its requirements.
<b>VROC_OFFLINE</b>	VROC Plugin is not working.
<b>INTERNAL_ERROR</b>	An unexpected error occurred during request process.

## Request Example:

```
PUT https://example.com/v1/vroc/config
{
  "locate_timeout": 12
}
```

```
{
  "data": null
  "status": {
    "success": true
  }
}
```

## 6.2 Clear metadata

```
[PUT] /v1/vroc/disks/:diskId/clear
```

Clears metadata from a disk.

### Request parameters

<b>:diskId</b>	string	Identifier of the disk.
----------------	--------	-------------------------

### Response parameters (Object)

<b>diskId</b>	string	Identifier of the cleared disk.
<b>usage</b>	string	Disk current usage. Should be equal to PASS_THROUGH.

### Available Errors

<b>INVALID_PARAMETER</b>	Unknown field is provided or diskId is invalid.
<b>OBJECT_NOT_FOUND</b>	DiskId points to not existing disk.
<b>NOT_SUPPORTED</b>	Provided enddevice is not a disk.
<b>VROC_OFFLINE</b>	VROC Plugin is not working.
<b>VROC_INVALID_STATE</b>	Disk is not available for clear metadata or is already cleared.
<b>INTERNAL_ERROR</b>	An unexpected error occurred during request process.

### Request Example:

```
PUT https://example.com/v1/vroc/disks/1183373271471488863/clear
null
```

```
{
  "data": {
    "diskId": "1183373271471488863",
    "usage": "PASS_THROUGH"
  },
  "status": {
    "success": true
  }
}
```

## 6.3 Mark/unmark disk as spare

```
[PUT] /v1/vroc/disks/:diskId/spare
```

Marks or unmarks disk as spare.

### Request parameters

<b>:diskId</b>	string	Identifier of the disk.
----------------	--------	-------------------------

<b>action</b>	string	Action to take. Possible values: MARK – marks disk as spare UNMARK – removes disk from spare state and makes it pass through
<b>Response parameters (Object)</b>		
<b>diskId</b>	string	Identifier of the cleared disk.
<b>usage</b>	string	Disk current usage. Should be equal to SPARE or PASS_THROUGH.
<b>Available Errors</b>		
<b>INVALID_PARAMETER</b>		Unknown field is provided or diskId is invalid or action parameter is invalid.
<b>OBJECT_NOT_FOUND</b>		DiskId points to not existing disk.
<b>NOT_SUPPORTED</b>		Provided disk is dual NVMe x8 disk drive.
<b>VROC_OFFLINE</b>		VROC Plugin is not working.
<b>VROC_INVALID_STATE</b>		Disk is not available to mark or unmark as spare.
<b>INTERNAL_ERROR</b>		An unexpected error occurred during request process.

## Request Example:

```
PUT https://example.com/v1/vroc/disks/1183373271471488863/spare
{
  "action": "MARK"
}
```

```
{
  "data": {
    "diskId": "1183373271471488863",
    "usage": "SPARE"
  },
  "status": {
    "success": true
  }
}
```

## 6.4 Mark disk as normal

<b>[PUT] /v1/vroc/disks/:diskId/marknormal</b>		
Marks failed disk as being healthy. Only supported on Windows.		
<b>Request parameters</b>		
<b>:diskId</b>	string	Identifier of the disk.
<b>Response parameters (Object)</b>		
<b>diskId</b>	string	Identifier of the cleared disk.
<b>state</b>	string	Disk current state. Should be equal to NORMAL.
<b>Available Errors</b>		
<b>INVALID_PARAMETER</b>		Unknown field is provided or diskId is invalid.

<b>OBJECT_NOT_FOUND</b>	DiskId points to not existing disk.
<b>NOT_SUPPORTED</b>	Functionality is not supported by VROC.
<b>VROC_OFFLINE</b>	VROC Plugin is not working.
<b>VROC_INVALID_STATE</b>	Disk is not available to mark as normal.
<b>INTERNAL_ERROR</b>	An unexpected error occurred during request process.

## Request Example:

```
PUT https://example.com/v1/vroc/disks/1183373271471488863/marknormal
null
```

```
{
  "data": {
    "diskId": "1183373271471488863",
    "state": "NORMAL"
  },
  "status": {
    "success": true
  }
}
```

## 6.5 Reset disk SMART event

```
[PUT] /v1/vroc/disks/:diskId/resetsmart
```

Resets SMART event on a disk. Only supported on Windows.

## Request parameters

<b>:diskId</b>	string	Identifier of the disk.
----------------	--------	-------------------------

## Response parameters (Object)

<b>diskId</b>	string	Identifier of the cleared disk.
<b>state</b>	string	Disk current state. Should be equal to NORMAL.

## Available Errors

<b>INVALID_PARAMETER</b>	Unknown field is provided or diskId is invalid.
<b>OBJECT_NOT_FOUND</b>	DiskId points to not existing disk.
<b>NOT_SUPPORTED</b>	Functionality is not supported by VROC.
<b>VROC_OFFLINE</b>	VROC Plugin is not working.
<b>VROC_INVALID_STATE</b>	Disk is not available to reset SMART event.
<b>INTERNAL_ERROR</b>	An unexpected error occurred during request process.

## Request Example:

```
PUT https://example.com/v1/vroc/disks/1183373271471488863/resetsmart
null
```

```
{
  "data": {
    "diskId": "1183373271471488863",
    "state": "NORMAL"
  },
  "status": {
    "success": true
  }
}
```

## 6.6 Managed Hotplug

[DELETE] /v1/vroc/disks/:diskId

Triggers hardware-specific illumination to locate the given disk and prepares it to be safely removed when OS is still running. It's only available for disks inside VMD controllers. There's no restriction for non-system disks and volumes. System disks cannot be removed. For system volumes, only 1 disk can be removed when status is normal (except RAID\_0 where all disks are not removable).

### Request parameters

<b>:diskId</b>	string	Identifier of the disk.
----------------	--------	-------------------------

### Response parameters (Object)

*null*

### Available Errors

<b>INVALID_PARAMETER</b>	Unknown field is provided or diskId is invalid.
<b>OBJECT_NOT_FOUND</b>	DiskId points to not existing disk.
<b>NOT_SUPPORTED</b>	Operation is not supported for specified disk.
<b>VROC_OFFLINE</b>	VROC Plugin is not working.
<b>VROC_INVALID_STATE</b>	Disk is not available to be flag as managed hotplug.
<b>INTERNAL_ERROR</b>	An unexpected error occurred during request process.

### Request Example:

```
DELETE https://example.com/v1/vroc/disks/1183373271471488863
null
```

```
{
  "data": null,
  "status": {
    "success": true
  }
}
```

## 6.7 Mark volume as normal

[PUT] /v1/vroc/volumes/:volumeId/marknormal		
Marks failed volume as being healthy. Only supported on Windows.		
Request parameters		
<b>:volumeId</b>	string	Identifier of the volume.
Response parameters (Object)		
<b>state</b>	string	Volume current state. Should be equal to NORMAL.
<b>volumeId</b>	string	Identifier of the cleared volume.
Available Errors		
<b>INVALID_PARAMETER</b>	Unknown field is provided or volumeId is invalid.	
<b>OBJECT_NOT_FOUND</b>	VolumeId points to not existing volume.	
<b>NOT_SUPPORTED</b>	Functionality is not supported by VROC.	
<b>VROC_OFFLINE</b>	VROC Plugin is not working.	
<b>VROC_INVALID_STATE</b>	Volume is not available to mark as normal.	
<b>INTERNAL_ERROR</b>	An unexpected error occurred during request process.	

### Request Example:

```
PUT https://example.com/v1/vroc/volumes/3251927784729082717/marknormal
null
```

```
{
  "data": {
    "state": "NORMAL",
    "volumeId": "3251927784729082717"
  },
  "status": {
    "success": true
  }
}
```

## 6.8 Rebuild volume

[PUT] /v1/vroc/volumes/:volumeId/rebuild		
Rebuilds degraded volume with a specified disk.		
Request parameters		
<b>:volumeId</b>	string	Identifier of the volume.
<b>diskId</b>	string	Identifier of the disk to add to rebuild.
Response parameters (Object)		
<b>state</b>	string	Volume current state. Should be equal to REBUILDING.
<b>volumeId</b>	string	Identifier of the cleared volume.
Available Errors		

<b>INVALID_PARAMETER</b>	Unknown field is provided or volumeId is invalid or diskId is invalid.
<b>OBJECT_NOT_FOUND</b>	VolumeId points to not existing volume or diskId points to not existing disk.
<b>NOT_SUPPORTED</b>	Operation is not supported by VROC. Disk is dual NVMe x8 drive or block sizes of a disk and volume are not equal or volume is not in degraded state.
<b>VROC_OFFLINE</b>	VROC Plugin is not working.
<b>VROC_INVALID_STATE</b>	Volume is not available to be rebuild.
<b>INTERNAL_ERROR</b>	An unexpected error occurred during request process.

## Request Example:

```
PUT https://example.com/v1/vroc/volumes/3251927784729082717/rebuild
{
  "diskId": "15573591210596179594"
}
```

```
{
  "data": {
    "state": "REBUILDING",
    "volumeId": "3251927784729082717"
  },
  "status": {
    "success": true
  }
}
```

## 6.9 Delete volume

<b>[DELETE] /v1/vroc/volumes/:volumeId</b>		
Deletes volume.		
Request parameters		
<b>:volumeId</b>	string	Identifier of the volume.
Response parameters (Object)		
<i>null</i>		
Available Errors		
<b>INVALID_PARAMETER</b>	Unknown field is provided or volumeId is invalid.	
<b>OBJECT_NOT_FOUND</b>	VolumeId points to not existing volume.	
<b>NOT_SUPPORTED</b>	Operation is not supported for specified volume.	
<b>VROC_OFFLINE</b>	VROC Plugin is not working.	
<b>VROC_INVALID_STATE</b>	Volume is not available to be removed.	
<b>INTERNAL_ERROR</b>	An unexpected error occurred during request process.	

## Request Example:

```
DELETE https://example.com/v1/vroc/volumes/3251927784729082717
null
```

```
{
  "data": null,
  "status": {
    "success": true
  }
}
```

## 6.10 Create volume

[POST] /v1/vroc/volumes

Creates volume from disks or in the array.

## Request parameters

<b>arrayId</b>	string	[Conditional parameter] Required when source is ARRAY. Represents identifier of the array to create a second volume.
<b>diskId</b>	string	[Optional parameter] Only valid when source is DISK. Represents an identifier of a single disk from disks array from which data must be migrated to the volume. Only supported on Windows.
<b>disks</b>	array	[Conditional parameter] Required when source is DISK. Represents an array of disks identifiers as strings to create a volume.
<b>journalingDriveId</b>	string	[Conditional parameter] Required when rwhPolicy is RWH_JOURNALING_DRIVE. Represents an identifier of a drive to be used as Journaling Drive.
<b>name</b>	string	Volume name. Must contain no more than 16 ASCII characters from range 32 to 126 (inclusive) without backslash (92). Spaces (20) are available but cannot be at the beginning. For Linux, also slash (47) is unavailable.
<b>raidLevel</b>	string	Name of RAID level to create with. Available values: RAID_0, RAID_1, RAID_5, RAID_10
<b>rwhPolicy</b>	string	[Optional parameter] Only valid for RAID_5. Represents RAID Write Hole policy for a new volume. Available values: RWH_OFF, RWH_DISTRIBUTED, RWH_JOURNALING_DRIVE. RWH_JOURNALING_DRIVE is only supported in Windows.
<b>size</b>	string	Size of a new volume in bytes. For maximum size, set size to 0. For second volume (source is ARRAY), size must be always maximum available so size also must be equal 0.
<b>source</b>	string	Determines source of a new volume. Available values: DISK - creates new array from provided disks and creates a first volume ARRAY - on existing array with one volume, creates a second volume

<b>stripSize</b>	string	[ <i>Optional parameter</i> ] Strip size to use for a new volume. If not provided, default strip size will be used (GET /v1/vroc/raids to get these values). For RAID 1, do not provide this field at all, it is not supported. Available values: SIZE_4KB, SIZE_8KB, SIZE_16KB, SIZE_32KB, SIZE_64KB, SIZE_128KB
<b>Response parameters (Object)</b>		
<b>arrayId</b>	string	Array identifier associated with the volume.
<b>arrayOrdinal</b>	integer	Position of the volume in the array.
<b>cachePolicy</b>	string	Current Cache Policy of the volume. This is not supported for Linux. Possible values: UNKNOWN, OFF, WRITE_THROUGH, WRITE_BACK, NOT_SUPPORTED
<b>id</b>	string	Volume identifier.
<b>initialized</b>	bool	If true, RAID mechanisms are fully initialized on the volume.
<b>journalingDriveId</b>	string	[ <i>Optional parameter</i> ] Identifier of the journaling drive associated to the volume. Only for RAID_5 and RWH Policy set to RWH_JOURNALING_DRIVE under Windows. This is not supported for Linux.
<b>migrProgress</b>	integer	Percentage progress for a running operation like migration, expansion or verify. Only valid if migration is true.
<b>migration</b>	bool	If true, the volume is during a operation and migrProgress shows meaningful data.
<b>name</b>	string	Volume name. Contains no more than 16 ASCII characters from range 32 to 126 (inclusive) without backslash (92). Spaces (20) are available but cannot be at the beginning. For Linux, also slash (47) is unavailable.
<b>numDisks</b>	integer	Number of disks used by the volume.
<b>raidLevel</b>	string	RAID level of the volume. Possible values: RAID_UNKNOWN, RAID_INVALID, RAID_0, RAID_1, RAID_5, RAID_10
<b>rwhPolicy</b>	string	[ <i>Optional parameter</i> ] Current RAID Write Hole policy. Only for RAID_5. Possible values: RWH_OFF, RWH_DISTRIBUTED, RWH_JOURNALING_DRIVE
<b>sectorSize</b>	object	Sector size of the volume.
<b>{}.logical</b>	string	Logical sector size in bytes.
<b>{}.physical</b>	string	Physical sector size in bytes.
<b>state</b>	string	Current volume state. Possible values: UNKNOWN, NORMAL, DEGRADED, FAILED, INITIALIZING, REBUILDING, VERIFYING, VERIFYING_AND_FIXING, GENERAL_MIGRATION, LOCKED, NON_REDUNDANT_VOLUME_FAILED_DISK

<b>stripSize</b>	string	Strip size of the volume. Possible values: SIZE_UNKNOWN, SIZE_4KB, SIZE_8KB, SIZE_16KB, SIZE_32KB, SIZE_64KB, SIZE_128KB
<b>systemVolume</b>	bool	If true, this volume holds OS.
<b>totalSize</b>	string	Total size of the volume in bytes.
<b>verify</b>	object	Last verify or verify and fix operation status.
<b>{}.badBlocks</b>	integer	Number of bad blocks found.
<b>{}.errors</b>	integer	Number of parity errors found.
<b>Available Errors</b>		
<b>INVALID_PARAMETER</b>		Unknown field is provided or any of fields are invalid or do not meet requirements of volume's RAID.
<b>OBJECT_NOT_FOUND</b>		ArrayId points to not existing array or diskId or journalingDriveId or any disk id in disks points to not existing disk or diskId is not one of provided disks.
<b>NOT_SUPPORTED</b>		RWH Policy is provided for not RAID 5 volume or any of provided disks mismatch with logical size.
<b>VROC_OFFLINE</b>		VROC Plugin is not working.
<b>VROC_INVALID_STATE</b>		Volume is not available to be created.
<b>INTERNAL_ERROR</b>		An unexpected error occurred during request process.

## Request Example:

```
POST https://example.com/v1/vroc/volumes
{
  "disks": [
    "15573591210596179594",
    "15296983155159336212"
  ],
  "name": "Volume1",
  "raidLevel": "RAID_0",
  "size": "107374182400",
  "source": "DISK",
  "stripSize": "SIZE_32KB"
}
```

```
{
  "data": {
    "arrayId": "14653709924164455516",
    "arrayOrdinal": 0,
    "cachePolicy": "NOT_SUPPORTED",
    "id": "2276443695438263509",
    "initialized": true,
    "migrProgress": 0,
    "migration": false,
    "name": "Volume1",
    "numDisks": 2,
    "raidLevel": "RAID_0",
    "sectorSize": {
      "logical": "512",
      "physical": "512"
    },
    "state": "NORMAL",
    "stripSize": "SIZE_32KB",
    "systemVolume": false,
    "totalSize": "107374182400",
    "verify": {
      "badBlocks": 0,
      "errors": 0
    }
  },
  "status": {
    "success": true
  }
}
```

#### Request Example:

```
POST https://example.com/v1/vroc/volumes
{
  "arrayId": "14653709924164455516",
  "name": "Volume2",
  "raidLevel": "RAID_1",
  "size": "0"
  "source": "ARRAY",
}
```

```

{
  "data": {
    "arrayId": "14653709924164455516",
    "arrayOrdinal": 1,
    "cachePolicy": "NOT_SUPPORTED",
    "id": "6689881952568389553",
    "initialized": true,
    "migrProgress": 0,
    "migration": true,
    "name": "Volume2",
    "numDisks": 2,
    "raidLevel": "RAID_1",
    "sectorSize": {
      "logical": "512",
      "physical": "512"
    },
    "state": "INITIALIZING",
    "stripSize": "N/A",
    "systemVolume": false,
    "totalSize": "346396033024",
    "verify": {
      "badBlocks": 0,
      "errors": 0
    }
  },
  "status": {
    "success": true
  }
}

```

## 6.11 Modify volume name

```
[PUT] /v1/vroc/volumes/:volumeId/name
```

Renames a specific volume's name.

### Request parameters

<b>:volumeId</b>	string	Identifier of the volume.
<b>name</b>	string	New name for the volume.

### Response parameters (Object)

<b>name</b>	string	Volume's current name. Should be equal to provided name. For Linux, there's a known issue that 16-length name will be cut to 15 characters.
<b>volumeId</b>	string	Identifier of the renamed volume. Can be different than previously.

### Available Errors

<b>INVALID_PARAMETER</b>	Unknown field is provided or volumeId is invalid or name is invalid, incorrect or already in use.
<b>OBJECT_NOT_FOUND</b>	VolumeId points to not existing volume.
<b>VROC_OFFLINE</b>	VROC Plugin is not working.
<b>VROC_INVALID_STATE</b>	Volume is not available to be renamed.
<b>INTERNAL_ERROR</b>	An unexpected error occurred during request process.

## Request Example:

```
PUT https://example.com/v1/vroc/volumes/3251927784729082717/name
{
  "name": "New_Volume"
}
```

```
{
  "data": {
    "name": "New_Volume",
    "volumeId": "7054548103759617029"
  },
  "status": {
    "success": true
  }
}
```

## 6.12 Modify volume size

```
[PUT] /v1/vroc/volumes/:volumeId/size
```

Resize a specific volume. For Windows, resize can be only modified to maximum. For Linux, only RAID 0 is not supported.

## Request parameters

<b>:volumeId</b>	string	Identifier of the volume.
<b>size</b>	string	New size in megabytes. To resize to maximum, it must be equal to 0.

## Response parameters (Object)

<b>size</b>	string	Volumes current size in megabytes.
<b>volumeId</b>	string	Identifier of the resized volume.

## Available Errors

<b>INVALID_PARAMETER</b>	Unknown field is provided or volumeId is invalid or size is invalid.
<b>OBJECT_NOT_FOUND</b>	VolumeId points to not existing volume.
<b>NOT_SUPPORTED</b>	Action is not supported.
<b>VROC_OFFLINE</b>	VROC Plugin is not working.
<b>VROC_INVALID_STATE</b>	Volume is not available to be resized.
<b>INTERNAL_ERROR</b>	An unexpected error occurred during request process.

## Request Example:

```
PUT https://example.com/v1/vroc/volumes/3251927784729082717/size
{
  "size": "153600"
}
```

```
{
  "data": {
    "size": "153600",
    "volumeId": "3251927784729082717"
  },
  "status": {
    "success": true
  }
}
```

### 6.13 Modify volume cache policy

[PUT] /v1/vroc/volumes/:volumeId/cachePolicy

Changes volume's cache policy. This is only supported on Windows.

#### Request parameters

<b>:volumeId</b>	string	Identifier of the volume.
<b>cachePolicy</b>	string	New cache policy to change. Possible values: OFF, WRITE_BACK

#### Response parameters (Object)

<b>cachePolicy</b>	string	Volumes current cache policy. Should be one of: OFF, WRITE_BACK
<b>volumeId</b>	string	Identifier of the volume with changed cache policy.

#### Available Errors

<b>INVALID_PARAMETER</b>	Unknown field is provided or volumeId is invalid or cachePolicy is invalid.
<b>OBJECT_NOT_FOUND</b>	VolumeId points to not existing volume.
<b>NOT_SUPPORTED</b>	Action is not supported.
<b>VROC_OFFLINE</b>	VROC Plugin is not working.
<b>VROC_INVALID_STATE</b>	Volume is not available to change its cache policy.
<b>INTERNAL_ERROR</b>	An unexpected error occurred during request process.

#### Request Example:

```
PUT https://example.com/v1/vroc/volumes/3251927784729082717/cachePolicy
{
  "cachePolicy": "WRITE_BACK"
}
```

```

{
  "data": {
    "cachePolicy": "WRITE_BACK",
    "volumeId": "3251927784729082717"
  },
  "status": {
    "success": true
  }
}

```

## 6.14 Modify volume RAID level

[PUT] /v1/vroc/volumes/:volumeId/level

Changes volume's RAID level and/or strip size.

### Request parameters

<b>:volumeId</b>	string	Identifier of the volume.
<b>disks</b>	array	[Optional parameter] Array of disks identifiers as strings to be used to change RAID level. It's required by some migrations to add more disks than there are in the volume, e.g. RAID 0 to RAID 5 requires 1 additional disk, from RAID 0 with 2 disks to RAID 10 requires 2 additional disks.
<b>raidLevel</b>	string	Name of the RAID level to change. Possible values: RAID_0, RAID_1, RAID_5, RAID_10
<b>stripSize</b>	string	[Optional parameter] Value of strip size to change. If not present, old strip size will be preserved. For RAID_1, this value cannot be provided. When changing to RAID_5 under VMD controller, omitting this field will change strip size to the most optimal for the number of disks.

### Response parameters (Object)

<b>disks</b>	array	[Optional parameter] If number of disks changed, this field is present and represents an array of disks identifiers as strings.
<b>raidLevel</b>	string	[Optional parameter] If raid level changed, this field is present and represents current RAID level.
<b>stripSize</b>	string	[Optional parameter] If strip size changed, this field is present and represents current strip size.
<b>volumeId</b>	string	Identifier of the volume with changed RAID level.

### Available Errors

<b>INVALID_PARAMETER</b>	Unknown field is provided or volumeId is invalid or any provided field is invalid.
<b>OBJECT_NOT_FOUND</b>	VolumeId points to not existing volume.
<b>NOT_SUPPORTED</b>	Action is not supported.
<b>VROC_OFFLINE</b>	VROC Plugin is not working.
<b>VROC_INVALID_RAID_LEVEL</b>	Provided RAID level is not compatible with disks inside volume or strip size was provided when changing to RAID 1.
<b>VROC_INVALID_STATE</b>	Volume is not available to change its level.

<b>INTERNAL_ERROR</b>	An unexpected error occurred during request process.
-----------------------	--

## Request Example:

```
PUT https://example.com/v1/vroc/volumes/8365939865289935182/level
{
  "disks": [
    "1183373271471488863"
  ],
  "raidLevel": "RAID_5",
  "stripSize": "SIZE_64KB"
}
```

```
{
  "data": {
    "disks": [
      "1282997796625283496",
      "1958265928247903799",
      "1183373271471488863"
    ],
    "raidLevel": "RAID_5",
    "stripSize": "SIZE_64KB",
    "volumeId": "8365939865289935182"
  },
  "status": {
    "success": true
  }
}
```

## 6.15 Modify volume RAID Write Hole

<b>[PUT] /v1/vroc/volumes/:volumeId/rwh</b>		
Changes RAID 5 volume's Write Hole policy.		
Request parameters		
<b>:volumeId</b>	string	Identifier of the volume.
<b>journalingDriveId</b>	string	[Optional parameter] Disk identifier to be associated with the volume as Journaling Drive. It is required only when rwhPolicy is equal to RWH_JOURNALING_DRIVE.
<b>rwhPolicy</b>	string	RAID Write Hole policy for the volume. Possible values: RWH_OFF, RWH_DISTRIBUTED, RWH_JOURNALING_DRIVE. RWH_JOURNALING_DRIVE is supported only on Windows.
Response parameters (Object)		
<b>journalingDriveId</b>	string	[Optional parameter] If present, Journaling Drive identifier was added to the volume.
<b>rwhPolicy</b>	string	RWH current policy. Should be one of: RWH_OFF, RWH_DISTRIBUTED, RWH_JOURNALING_DRIVE.
<b>volumeId</b>	string	Identifier of the volume with changed RWH policy.

Available Errors	
<b>INVALID_PARAMETER</b>	Unknown field is provided or volumeId is invalid or rwhPolicy is invalid or journalingDriveId is invalid or missing when required.
<b>OBJECT_NOT_FOUND</b>	VolumeId points to not existing volume or journalingDriveId points to not existing disk.
<b>NOT_SUPPORTED</b>	Action is not supported or journaling drive has block size mismatch with volume disks.
<b>VROC_OFFLINE</b>	VROC Plugin is not working.
<b>VROC_INVALID_STATE</b>	Volume is not available to change its RWH policy or volume is not RAID 5.
<b>INTERNAL_ERROR</b>	An unexpected error occurred during request process.

## Request Example:

```
PUT https://example.com/v1/vroc/volumes/8365939865289935182/cachePolicy
{
  "journalingDriveId": "220729753440415355",
  "rwhPolicy": "RWH_JOURNALING_DRIVE"
}
```

```
{
  "data": {
    "journalingDriveId": "220729753440415355",
    "rwhPolicy": "RWH_JOURNALING_DRIVE",
    "volumeId": "8365939865289935182"
  },
  "status": {
    "success": true
  }
}
```

## 6.16 Initialize volume

```
[PUT] /v1/vroc/volumes/:volumeId/init
```

Initializes the process of data redundancy for volume's RAID level. Only supported on Windows as Linux initializes automatically after creation. It is not supported for RAID 0 because it does not require any initialization.

## Request parameters

<b>:volumeId</b>	string	Identifier of the volume.
------------------	--------	---------------------------

## Response parameters (Object)

<b>state</b>	string	Volume current state. Should be equal to INITIALIZING.
<b>volumeId</b>	string	Identifier of the initialized volume.

## Available Errors

<b>INVALID_PARAMETER</b>	Unknown field is provided or volumeId is invalid.
<b>OBJECT_NOT_FOUND</b>	VolumeId points to not existing volume.

<b>NOT_SUPPORTED</b>	Functionality is not supported by VROC.
<b>VROC_OFFLINE</b>	VROC Plugin is not working.
<b>VROC_INVALID_STATE</b>	Volume is not available to initialize.
<b>INTERNAL_ERROR</b>	An unexpected error occurred during request process.

## Request Example:

```
PUT https://example.com/v1/vroc/volumes/8355417539050691081/init
null
```

```
{
  "data": {
    "state": "INITIALIZING",
    "volumeId": "8355417539050691081"
  },
  "status": {
    "success": true
  }
}
```

## 6.17 Start/Cancel volume verification

```
[PUT] /v1/vroc/volumes/:volumeId/verify
```

Starts or cancels a volume verify operation on the volume. This is not supported for RAID 0.

## Request parameters

<b>:volumeId</b>	string	Identifier of the volume.
<b>action</b>	string	Action to take. Possible values: START - starts verification CANCEL - cancels verification
<b>repair</b>	bool	[Optional parameter] Determines if volume should also resolve conflicts of redundancy. Parameter is required only when action is equal to START.

## Response parameters (Object)

<b>state</b>	string	Volume current state. Should be equal to VERIFYING or VERIFYING_AND_FIXING.
<b>volumeId</b>	string	Identifier of the verifying volume.

## Available Errors

<b>INVALID_PARAMETER</b>	Unknown field is provided or volumeId is invalid or action parameter is invalid or repair is missing when required.
<b>OBJECT_NOT_FOUND</b>	VolumeId points to not existing volume.
<b>NOT_SUPPORTED</b>	Action is not supported.
<b>VROC_OFFLINE</b>	VROC Plugin is not working.
<b>VROC_INVALID_STATE</b>	Volume is not available to verify.
<b>INTERNAL_ERROR</b>	An unexpected error occurred during request process.

## Request Example:

```
PUT https://example.com/v1/vroc/volumes/4810375228359037707/verify
{
  "action": "VERIFY",
  "repair": true
}
```

```
{
  "data": {
    "state": "VERIFYING_AND_FIXING",
    "volumeId": "4810375228359037707"
  },
  "status": {
    "success": true
  }
}
```

## 6.18 Add disks to array

```
[PUT] /v1/vroc/arrays/:arrayId
```

Adds one or more disks to the array that does not have RAID 1 or RAID 10.

## Request parameters

<b>:arrayId</b>	string	Identifier of the array.
<b>disks</b>	array	Array of disks identifiers as strings to used them to resize the array.

## Response parameters (Object)

<b>arrayId</b>	string	Identifier of the modified array.
<b>disks</b>	array	Array of disks identifiers as strings that are currently in the array.

## Available Errors

<b>INVALID_PARAMETER</b>	Unknown field is provided or arrayId is invalid or disks is invalid or empty or any of its identifiers are invalid.
<b>OBJECT_NOT_FOUND</b>	ArrayId points to not existing array or any disk id points to not existing disk.
<b>VROC_OFFLINE</b>	VROC Plugin is not working.
<b>VROC_INVALID_STATE</b>	Array is busy.
<b>INTERNAL_ERROR</b>	An unexpected error occurred during request process.

## Request Example:

```
PUT https://example.com/v1/vroc/arrays/3343385868806453688
{
  "disks": [
    "6029574419765821177"
  ]
}
```

```
{
  "data": {
    "arrayId": "3343385868806453688",
    "disks": [
      "6029574419765821177",
      "10226389754048326290",
      "9221216980796389618",
      "8628482652860055311"
    ]
  },
  "status": {
    "success": true
  }
}
```

## 6.19 Set array write cache state

[PUT] /v1/vroc/arrays/:arrayId/writecache

Sets the write cache state for the array. This is only supported on Windows.

### Request parameters

<b>:arrayId</b>	string	Identifier of the array.
<b>state</b>	string	State of the write cache. Available values: ENABLED - enables disk cache for all disks in the array DISABLED - disables disk cache for all disks in the array

### Response parameters (Object)

<b>arrayId</b>	string	Identifier of the modified array.
<b>writeCachePolicy</b>	string	[Optional parameter] If Write Cache Policy changed, this field is present and represents current policy state. Possible value: ON, OFF

### Available Errors

<b>INVALID_PARAMETER</b>	Unknown field is provided or arrayId is invalid or state is invalid.
<b>OBJECT_NOT_FOUND</b>	ArrayId points to not existing array.
<b>VROC_OFFLINE</b>	VROC Plugin is not working.
<b>VROC_INVALID_STATE</b>	Array is busy.
<b>INTERNAL_ERROR</b>	An unexpected error occurred during request process.

### Request Example:

```
PUT https://example.com/v1/vroc/arrays/3343385868806453688/writecache
{
  "state": "ENABLED"
}
```

```
{
  "data": {
    "arrayId": "3343385868806453688",
    "state": "ON"
  },
  "status": {
    "success": true
  }
}
```

## 6.20 Port locate

[PUT] /v1/vroc/ports/:portId/locate

Triggers hardware-specific illumination to locate the given port.

### Request parameters

<b>:portId</b>	string	Identifier of the port.
<b>state</b>	string	State of locate on the port. Available values: ENABLED - starts the illumination for specific period of time DISABLED - stops the illumination. Only available for Linux

### Response parameters (Object)

*null*

### Available Errors

<b>INVALID_PARAMETER</b>	Unknown field is provided or portId is invalid or state is invalid.
<b>OBJECT_NOT_FOUND</b>	PortId points to not existing port.
<b>NOT_SUPPORTED</b>	Action is not supported.
<b>VROC_OFFLINE</b>	VROC Plugin is not working.
<b>INTERNAL_ERROR</b>	An unexpected error occurred during request process.

### Request Example:

```
PUT https://example.com/v1/vroc/ports/2303482965723205661/locate
{
  "state": "ENABLED"
}
```

```
{
  "data": null,
  "status": {
    "success": true
  }
}
```

## 6.21 Rescan for hardware changes

[POST] /v1/vroc/rescan	
Forces the driver and/or system to scan for hardware changes. This is only supported on Windows.	
Request parameters	
<i>null</i>	
Response parameters (Object)	
<i>null</i>	
Available Errors	
<b>INVALID_PARAMETER</b>	Unknown field is provided.
<b>NOT_SUPPORTED</b>	Action is not supported.
<b>VROC_OFFLINE</b>	VROC Plugin is not working.
<b>INTERNAL_ERROR</b>	An unexpected error occurred during request process.

### Request Example:

```
POST https://example.com/v1/vroc/rescan
null
```

```
{
  "data": null,
  "status": {
    "success": true
  }
}
```

## 6.22 Read patrol

[PUT] /v1/vroc/controllers/:controllerId/readpatrol		
Sets the read patrol state on the controller. This is only supported on Windows.		
Request parameters		
<b>:controllerId</b>	string	Identifier of the controller.
<b>state</b>	string	State of read patrol on the controller. Available values: ENABLED - enables read patrol DISABLED - disables read patrol
Response parameters (Object)		
<b>controllerId</b>	string	Identifier of the controller.
<b>state</b>	string	Current state of the controller's read patrol. Possible values: ENABLED, DISABLED
Available Errors		

<b>INVALID_PARAMETER</b>	Unknown field is provided or controllerId is invalid or state is invalid.
<b>OBJECT_NOT_FOUND</b>	ControllerId points to not existing controller.
<b>NOT_SUPPORTED</b>	Controller does not support read patrol.
<b>VROC_OFFLINE</b>	VROC Plugin is not working.
<b>VROC_INVALID_STATE</b>	Controller is not available to change read patrol.
<b>INTERNAL_ERROR</b>	An unexpected error occurred during request process.

## Request Example:

```
PUT https://example.com/v1/vroc/controllers/923094678399631677/readpatrol
{
  "state": "ENABLED"
}
```

```
{
  "data": {
    "controllerId": "923094678399631677",
    "state": "ENABLED"
  },
  "status": {
    "success": true
  }
}
```

## 6.23 Rebuild on hot insert

```
[PUT] /v1/vroc/controllers/:controllerId/rohi
```

Sets the rebuild on hot insert state on the controller. This is only supported on Windows.

## Request parameters

<b>:controllerId</b>	string	Identifier of the controller.
<b>state</b>	string	State of rohi on the controller. Available values: ENABLED - enables rohi DISABLED - disables rohi

## Response parameters (Object)

<b>controllerId</b>	string	Identifier of the controller.
<b>state</b>	string	Current state of the controller's rohi. Possible values: ENABLED, DISABLED

## Available Errors

<b>INVALID_PARAMETER</b>	Unknown field is provided or controllerId is invalid or state is invalid.
<b>OBJECT_NOT_FOUND</b>	ControllerId points to not existing controller.

<b>NOT_SUPPORTED</b>	Controller does not support rohi.
<b>VROC_OFFLINE</b>	VROC Plugin is not working.
<b>VROC_INVALID_STATE</b>	Controller is not available to change rohi.
<b>INTERNAL_ERROR</b>	An unexpected error occurred during request process.

### Request Example:

```
PUT https://example.com/v1/vroc/controllers/923094678399631677/rohi
{
  "state": "ENABLED"
}
```

```
{
  "data": {
    "controllerId": "923094678399631677",
    "state": "ENABLED"
  },
  "status": {
    "success": true
  }
}
```