REST API 1.13

This guide describes how to use the Pure Storage REST API. The guide is divided into two sections: Overview and Resources. The Overview section describes high-level concepts while the Resources section contains a comprehensive list of resources and endpoints.

REST API Overview

The Pure Storage REST API provides simple interfaces for many Purity commands.

What is REST?

REpresentational State Transfer (REST) uses HTTP requests to interact with resources within Pure Storage. REST is an architectural style that describes the following constraints:

- **Uniform Interface**. Individual resources are identified in requests using URIs as resource identifiers. For example:

```
volume
volume/v1
```

Clients manipulate resources through request methods, such as:

Method	Description	Example
GET	List resources.	`GET volume`
POST	Create a resource.	`POST volume/v1`
PUT	Modify a resource.	`PUT volume/v1`
DELETE	Delete a resource.	`DELETE volume/v1`

Requests can include additional parameters, either in the URL or as JSON post data. For example, the `snap` parameter is included in the following request to list all snapshots:

```
GET volume?snap=true
```

Responses return a response code and data (represented in JSON). For example, the following `GET volume/v1` request returns a response code `200 OK` and the attribute details for a volume named "v1":

```
>> GET volume/v1

<< 200 OK
  {
    "created": "Tue Jan 28 06:15:23 2014",
    "name": "v1",
    "serial": "CEE061D3B3B23499000101E1",
    "size": 1073741824,
    "source": null
  }
```

- **Stateless**. The necessary state to handle a request is contained in the request itself.
- **Cacheable**. The responses must implicitly or explicitly define themselves as cacheable.
- **Client-Server**. The uniform interface creates separation between client and server. Clients can be more portable, severs do not need to know anything about user interface or state.
- **Layered System**. A client cannot tell if it is connected directly to the server; intermediary servers can be used to improve scalability or enforce security policies.

Pure Storage REST API

The Pure Storage REST API provides simple interfaces for many Purity commands.

```
purearray list
```

In the following example, the Purity CLI command returns the attributes for the array:

```
$ purearray list
Name      ID                                    Version  Revision
pure-001  9d2c3325-c5b9-14ad-43b8-1c773a6406b90  4.8.0    201510290821+a1af0ba
```

The equivalent REST API request `GET https://pure-001.example.com/api/1.13/array` produces the same results:

```
>> GET https://pure-001.example.com/api/1.13/array

<< 200 OK

{
  "array_name": "pure-001",
  "id": "9d2c3325-c5b9-14ad-43b8-1c773a6406b9",
  "revision": "201510290821+a1af0ba",
  "version": "4.8.0"
}
```

Using the REST API consists of the following steps:

1. Generate the API token. This is a one-time step.
2. Create the REST session.
3. Submit REST API requests.
4. Invalidate the REST session.

Generating an API Token

The REST API uses authentication tokens to create sessions. All Purity users can generate their own API tokens.

Generate the API token through the Purity GUI (System > Users > Create API Token) or CLI (`pureadmin create --api-token`).

API tokens can also be generated through the REST API. Generating API tokens through the REST API may be convenient during development, but is discouraged in production environments as it exposes usernames and passwords.

Once generated, the API token is valid until it is deleted or recreated.

Creating a REST Session

After the API token has been generated, use the token to create a REST session.

To create a REST session, send the following request and payload:

```
>> POST https://pure-001.example.com/api/1.13/auth/session

<< 200 OK

{
  "api_token": "a00bcd9e-f3gh-20i7-j51k-70l3515mn300"
}
```

The session is valid until it is explicitly deleted (invalidated) or experiences 30 minutes of inactivity.

Submitting REST API Requests

REST API requests can be performed within a REST session.

The user's authorization role in Purity determines which requests the client can perform within the session.

Each REST API request is comprised of a complete URL. The complete URL used to make a Pure Storage REST request includes the following components:

- Method (GET, POST, PUT, or DELETE)
- Purity array
- REST API version
- URI

For example, the following request lists all volumes and their attributes:

```
GET https://pure-001.example.com/api/1.13/volume
```

Some REST API requests include additional parameters.

Parameters for GET methods may be specified as query parameters in the URL. For example, the following request, with `space` specified as a query parameter, lists array space usage:

```
GET https://pure-001.example.com/api/1.13/array?space=true
```

All other methods (POST, PUT, and DELETE) must be specified as JSON post data. Refer to the http://www.json.org/ website for more information about the JSON interchange format.

The following are examples of how different data types are formatted in JSON:

Data Type	Examples
boolean	`{"eradicate": true}`
object	`{"replicate_blackout": {"end": 14400,"start": 7200}}`
list	`{"nameservers": ["192.168.1.1","192.168.0.1"]}`
number	`{"per_day": 12}`
string	`{"domain": "example.com"}`

For example the following request and payload, with the `source` parameter specified as JSON post data, creates a new volume named v6 by copying volume v1.

```
>> POST https://pure-001.example.com/api/1.13/volume/v6

<< 200 OK

{
  "source": "v1"
}
```

Refer to Resources for a list of all available Pure Storage REST requests.

Each REST API request returns a response code. Successful requests return the response code `200 OK`. Scripts must be prepared to handle responses with codes other than `200 OK`.

In the following example, the client is attempting to list alert information for email recipient `support_@purestorage.com`. The request returns a response code `400 BAD REQUEST`, stating that recipient `support_@purestorage.com` does not exist.

```
>> GET http://lt14/api/1.13/alert/support_@purestorage.com

<< 400 BAD REQUEST
{
    msg: "Address does not exist."
    ctx: "support_@purestorage.com"
}
```

Refer to the Response Codes table for a description of each response code.

Every request that modifies a resource is logged in the Purity audit trail. These requests include most POST, PUT, and DELETE requests. The logged events can be viewed through the Purity GUI (Messages > Audit Trail) and CLI (`puremessage list --audit`).

Invalidating a REST Session

A REST session is valid until it is explicitly deleted (invalidated) or experiences 30 minutes of inactivity.

To manually invalidate a REST session, send the following request:

```
>> DELETE https://{array}/api/1.13/auth/session

<< 200 OK
{
  "username": "pureuser"
}
```

Version Control

The REST API consists of a major and minor version.

Major releases modify or remove existing functionality and are not backward compatible.

Minor releases add new functionality only. Minor releases are backward compatible, with exceptions listed in the REST API release notes. For example, users on REST API version 1.2 can submit any of the requests available in REST API versions 1.0 through 1.2.

To determine the supported REST API versions for the current release, enter the URL `https://{array}/api/api_version`. The supported version numbers are returned as a list. For example,

```
{"version": ["1.0", "1.1", "1.2", "1.3", "1.4", "1.5", "1.6", "1.7", "1.8", "1.9", "1.10", "1.11"]}
```

Authentication vs. Authorization

The REST API uses authentication tokens instead of credentials to create sessions, while authorization roles determine which requests can be performed within a session.

Authentication

Purity clients must use an API token in order to securely create REST API sessions. By using API tokens to authenticate, scripts do not have to expose usernames and passwords. The Purity username and password are required initially to create the API token.

API tokens are unique to the Purity user for whom it was created. Each Purity user can generate and view only their own API token. Once created, an API token is valid until it is deleted or recreated.

API tokens can be created and recreated through the Purity GUI (System > Users > Create API Token), CLI (`pureadmin create --api-token`) or REST API.

After obtaining an API token, clients can create REST API sessions and start sending requests. Since REST API service sessions are completely separate from Purity GUI sessions, REST requests cannot be accessed through the Purity GUI.

API token management does not affect Purity usernames and passwords. For example, invalidating an API token does not invalidate the Purity username or password that was used to create the token. Likewise, changing the Purity username or password does not affect the API token.

Authorization

Authorization determines what the client is permitted to do within a REST session.

If Purity is integrated with a directory service, such as Microsoft Active Directory or OpenLDAP, the REST service leverages role-based access control (RBAC) to determine permission levels. In other words, the roles that are assigned to the users in a directory service are the same roles that are used in Purity and in the REST API.

Once you have created a REST API session, the requests that the client is authorized to make depend on the role of the user associated with the API token (as system administrator, storage administrator, or read-only user).

The integration of RBAC in the REST API with directory services is dynamic. The API token permissions are integrated with the Active Directory permissions. Changes made to a user's Active Directory permissions affect the associated API token permissions.

For example, if a user is moved to a different directory services group and the permission level changes from read-only user to storage administrator, the associated API token permissions change as well. Permission changes made in the directory server are cached on the array and may take 30 minutes to affect the API token.

Response Codes

The Pure Storage REST API returns the following HTTP response status codes for the GET, POST, PUT, and DELETE operations:

Code	Description
200 OK	The request is successful.
400 BAD REQUEST	The request includes an invalid action or invalid or missing data. Refer to the REST API response for more information.
401 UNAUTHORIZED	The REST API session has not been created or has expired. Create a new session.
403 FORBIDDEN	The request is valid, but the client is not authorized to perform the action. For example, a read-only client is not allowed to create volumes.
404 NOT FOUND	The URI is not valid.
405 METHOD NOT ALLOWED	The method is not valid for the URI specified. Enter a GET, POST, PUT, or DELETE method that is valid for the URI specified.

450 VERSION NOT SUPPORTED	The request is not valid for the REST API version specified.
500 SERVER ERROR	The request generated an internal error. Contact Pure Storage Support for help.

Sorting, Filtering, and Pagination

When submitting GET requests that return a large amount of data, include filtering, sorting, and pagination query parameters to help you better manage the results.

The Pure Storage REST API supports filtering, sorting, and pagination for the following GET methods: GET volume, GET host, GET hgroup, GET pgroup, and GET port.

Sorting

The sort query parameter returns the response objects of a GET request in the order specified. Sorting can be performed on any of the fields in the response, and the response objects can be sorted in ascending or descending order.

Syntax

`?sort=`**`field_name`**

`field_name`

Name of the field to sort. `Field_name` can be any of the fields in the response. To sort in descending order, append the minus sign (-) to the field name. To sort on multiple field names, list the field names as comma-separated values.

Examples

Example 1: Get a list of all protection groups sorted in ascending order by protection group name.

```
GET https://pure01.example.com/api/1.13/pgroup?sort=name
```

Example 2: Get a list of all host groups sorted in descending order by host group name.

```
GET https://pure01.example.com/api/1.13/hgroup?sort=name-
```

Example 3: Get a list of all volumes sorted in descending order by volume size, and then sorted in ascending order by volume name.

```
GET https://pure01.example.com/api/1.13/volume?sort=size-,name
```

Filtering

The filter query parameter narrows down the results of a GET request to only the response objects that satisfy the filter criteria.

Filtering With Operators

Pure Storage REST API filter queries support common comparison operators.

Syntax

`?filter=`**`field_name operator field_value`**

`field_name`

Name of the field on which to create the filter. `Field_name` can be any of the fields in the response.

`operator`

Type of filter match used to compare `field_name` to `field_value`. Filters support the following operators:

Operator	Description	Examples
=	Equals	Get a list of all volumes with volume names that include `v1`. `/volume?filter=name='*v1*'`
!=	Does not equal	Get a list of all volumes with volume names that do not include `v1`. `/volume?filter=name!='*v1*'`
<	Less than	Get a list of all volumes that are less than 100 gigabytes in size.

		`/volume?filter=size<107374182400`

`>`	Greater than	Get a list of all volumes that are greater than 100 gigabytes in size.
		`/volume?filter=size>107374182400`
		Get a list of all volumes that were created after 2016-04-25T23:02:23Z.
		`/volume?filter=created>'2016-04-25T23:02:23Z'`

`<=`	Less than or equal to	Get a list of all volumes that are less than or equal to 100 gigabytes in size.
		`/volume?filter=size<=107374182400`
		Get a list of all volumes that were created on or before 2016-04-25T23:02:23Z.
		`/volume?filter=created<='2016-04-25T23:02:23Z'`

`>=`	Greater than or equal to	Get a list of all volumes that are greater than or equal to 100 gigabytes in size. `/volume?filter=size>=107374182400`

`field_value`

Search value (number, date, or string) that determines the fields to be included in or excluded from the response.

Literal strings must be wrapped in quotes.

To refine the filter, include the asterisk (*) symbol as a wildcard character in the field value to represent zero or more characters. `Field_value` can include multiple asterisks, and the asterisks can appear anywhere in the value. For example, the request `GET /host?filter=name='*cat*'` lists all hosts with names that contain `cat`, including ones that begin or end with `cat`, such as `cat`, `catnap`, `happycats`, and `lolcat`. If the asterisks were not included, only the host named cat would be returned.

Examples

Example 1: Get the volume with serial number `31F5207543354311000 1103F`.

```
GET https://pure01.example.com/api/1.13/volume?filter=serial='31F5207543354311000 1103F'
```

Example 2: Get a list of hosts that are associated with host group `ESXi-IT-Cluster02-hg2`.

```
GET https://pure01.example.com/api/1.13/host?filter=hgroup='ESXi-IT-Cluster02-hg2'
```

Example 3: Get a list of all Fibre Channel ports on controller `ct0`.

```
GET https://pure01.example.com/api/1.13/port?filter=name='ct0.fc*'
```

Example 4: Get a list of all volumes that have been destroyed and will be eradicated within the next 2 hours (7200 seconds).

```
GET https://pure01.example.com/api/1.13/volume?pending_only=true&filter=time_remaining<='7200'
```

Filtering With Functions

Pure Storage REST API filter queries support the CONTAINS and NOT functions.

Syntax

`?filter=`**`function(parameters)`**

Filters support the following functions:

Function	Description	Examples
`contains(field_name,string)`	Contains the enclosed string. Takes exactly two parameters, where `field_name` represents the name of the field on which to create the filter, and `string` represents the string to search within `field_name`.	Get a list of all host groups with names that include `cluster02`. `/host?filter=contains(hgroup,'cluster02')`
`not(expression)`	Inverse of the enclosed expression.	Get a list of all host groups with names that do not

		include `cluster02`. `/host?filter=not(hgroup='cluster02')`

Examples

Example 1: Get a list of all hosts that are associated with host groups that include `hg3` in the host group name.

```
GET https://pure01.example.com/api/1.13/host?filter=contains(hgroup,'hg3')
```

Example 2: Get a list of all volumes with names that do not include `vol01`.

```
GET https://pure01.example.com/api/1.13/volume?filter=not(contains(name,'vol01'))
```

Existence Checks

The REST API supports existence checks. For example, the request `GET /volume?filter=name` checks to see if "name" field name exists in the volume resource.

Examples

Example 1: Get a list of all hosts that are associated with a host group.

```
GET https://pure01.example.com/api/1.13/host?filter=hgroup
```

Example 2: Get a list of all volumes that were not created from another source.

```
GET https://pure01.example.com/api/1.13/volume?filter=not(source)
```

Example 3: Get a list of all hosts that are not associated with Fibre Channel WWNs. (Sample response included.)

```
>>  GET https://pure01.example.com/api/1.13/host?filter=not(wwn)

<<  {
      "iqn": [],
      "wwn": [],
      "name": "ESXi-Cluster02-v7",
      "hgroup": null
    },
    {
      "iqn": [],
      "wwn": [],
      "name": "ESXi-Cluster02-v8",
      "hgroup": "ESXHGroup1"
    },
    {
      "iqn": [],
      "wwn": [],
      "name": "ESXi-Cluster02-v9",
      "hgroup": null
    },
    {
      "iqn": [],
      "wwn": [],
      "name": "ESXi-Cluster02-v10",
      "hgroup": null
    }
```

Example 4: Get a list of all protection groups that do not contain volume or host group members.

```
GET https://pure01.example.com/api/1.13/pgroup?filter=not(volumes) and not(hgroups)
```

Pagination

Pagination controls the amount of data returned by displaying response objects (items) as page-by-page lists.

Pagination only applies if the total number of items returned is greater than the limit specified. To determine the total number of items returned, see the `x-total-item-count` header field of the response.

The Pure Storage REST API offers two methods of pagination: page by index, and page by continuation token.

Page by Index

Page by index enables clients to control the size and starting point of a page. Use page by index, such as in GUI implementations, to navigate through response objects page by page, jump to specific pages, and jump to the first or last page. For example, to page through all results with 20 results per page, begin with `start=0&limit=20`.

With page by index pagination, consistency may not be guaranteed in that some data may be duplicated or missing across pages. If consistency is important, use continuation tokens.

Syntax

`start=`**`starting_index`**`&limit=`**`items_returned`**

`starting_index`

Index number of the first item you want results for.

`items_returned`

Maximum number of response objects returned per page.

Examples

Example 1: Get all volumes starting at the very first volume.

```
GET https://pure01.example.com/api/1.13/volume?start=0
```

Example 2: Get 10 volumes at a time.

```
GET https://pure01.example.com/api/1.13/volume?limit=10
```

Example 3: Get 10 volumes with index 0 to 9.

```
GET https://pure01.example.com/api/1.13/volume?start=0&limit=10
```

Example 4: Get 1 host with index 100.

```
GET https://pure01.example.com/api/1.13/hosts?start=100&limit=1
```

Page by Continuation Token

The REST API supports continuation tokens, giving clients the ability to paginate with some consistency guaranteed.

Page by continuation token requires the `limit` and `token` query parameters.

The `limit` parameter determines how many response objects (items) are returned on each page.

The `token` parameter is a Base64 encoded value that is used to get the next page of data. The token is taken from the `x-next-token` header field of the response. A query has reached its last page when the response does not include a token.

Syntax

`limit=`**`items_returned`**`&token=`**`token`**

`items_returned`

Maximum number of response objects returned per page.

`token`

Token of the last/first item returned in the current paginated request. The token is used in the next submit to determine the first item that should appear on the next page. The token appears in the response header. To ensure consistent paging, the `token` parameter must always be used with the `sort` parameter.

Examples

Example 1: Use token `aWQgPSA5ODA1Mg==` to get the next 10 volumes.

```
GET https://pure01.example.com/api/1.13/volume?limit=10&token=aWQgPSA5ODA1Mg==
```

Example 2: Use token `c3JjX2FwdCA9IDY2LGlkID0gMQ==` to get the next 10 protection groups.

```
GET https://pure01.example.com/api/1.13/pgroup?limit=10&token=c3JjX2FwdCA9IDY2LGlkID0gMQ==
```

Example 3: Get 2 volumes with index 5 to 9, and then use the token returned (aWQgPSA2OTY1NA==) to get the next 2 volumes.

```
>>  GET https://pure01.example.com/api/1.13/volume?start=5&limit=2

<<  x-next-token → aWQgPSA2OTY1NA==
    x-total-item-count →21

<<  {
      "source": null,
      "serial": "122E64ABC3FE4BA100011015",
      "created": "2016-04-28T20:30:56Z",
      "name": "ESXi-Cluster02-vol005",
      "size": 107374182400
    },
    {
      "source": null,
      "serial": "122E64ABC3FE4BA100011016",
      "created": "2016-04-28T20:30:56Z",
      "name": "ESXi-Cluster02-vol007",
      "size": 107374182400
    }

>>  GET https://pure01.example.com/api/1.13/volume?limit=2&token=aWQgPSA2OTY1NA==

<<  x-next-token → aWQgPSA2OTY2MQ==
<<  {
      "source": null,
      "serial": "122E64ABC3FE4BA10001101C",
      "created": "2016-04-28T20:30:56Z",
      "name": "ESXi-Cluster02-vol006",
      "size": 107374182400
    },
    {
      "source": null,
      "serial": "122E64ABC3FE4BA10001101D",
      "created": "2016-04-28T20:30:56Z",
      "name": "ESXi-Cluster01-vol001",
      "size": 107374182400
    }
```

Example 4: Get one protection group at a time using the token returned each time to get the next protection group.

```
>>  GET https://pure01.example.com/api/1.13/pgroup?limit=1

<<  x-next-token → c3JjX2FwdCA9IDY2LGlkID0gMQ==
    x-total-item-count →10
<<  {
      "name": "PG001",
      "hgroups": null,
      "source": "pure001",
      "hosts": [
        "ESXi-PROD-Cluster01-h2",
        "ESXi-STG-Cluster03-h2"
      ],
      "volumes": null,
      "targets": null
    }

>>  GET https://pure01.example.com/api/1.13/pgroup?limit=1&token=c3JjX2FwdCA9IDY2LGlkID0gMQ==

<<  x-next-token → c3JjX2FwdCA9IDY2LGlkID0gMg==
<<  {
      "name": "PG002",
      "hgroups": null,
      "source": "pure001",
      "hosts": [
        "ESXi-GRP-Cluster02-h2",
        "ESXi-GRP-Cluster02-h3"
      ],
      "volumes": null,
      "targets": null
    }

>>  GET https://pure01.example.com/api/1.13/volume?limit=1&token=c3JjX2FwdCA9IDY2LGlkID0gMg==

<<  x-next-token → c3JjX2FwdCA9IDY2LGlkID0gMw==
<<  {
      "name": "PG003",
      "hgroups": null,
      "source": "pure005",
      "hosts": null,
      "volumes": [
        "ESXi-Cluster01-vol007",
        "ESXi-Cluster01-vol010",
        "ESXi-Cluster01-vol009",
        "ESXi-Cluster01-vol008"
      ],
      "targets": null
    }
```

Combining Sorting, Filtering, and Pagination

Follow these guidelines when combining sorting, filtering, and pagination:

- Sort queries can be combined. A single request can be sorted on multiple field names. For example, you can sort all volumes from largest to smallest volume size, and then sort volumes of the same size in ascending order by volume name. To sort on multiple field names, list the field names as comma-separated values.
- Filter queries can be combined using the AND and OR operators. The AND operator displays response objects that meet the criteria of all filter queries in the request. The OR operator displays response objects that meet the criteria of at least one of the filter queries in the request.
- The `sort`, `filter`, `limit`, and `start` filters can all be combined together.

Examples

Example 1: Get a list of all volumes sorted in descending order by volume size, and then sorted in ascending order by volume name.

```
GET https://pure01.example.com/api/1.13/volume?sort=size-,name
```

Example 2: Get a list of all hosts with names that include `h2` and are associated with host groups with names that include `hg3`.

```
GET https://pure01.example.com/api/1.13/host?filter=(contains(name,'h2') and contains(hgroup,'hg3'))
```

Example 3: Get a list of all volumes that are greater than 10 gigabytes in size and were created after 2016-04-25T23:02:23Z.

```
GET https://pure01.example.com/api/1.13/volume?filter=size>10737418240 and created>'2016-04-25T23:02:23Z'
```

Example 4: Get a list of all volumes that begin with `v1`, and sort the list in descending order by name.

```
GET https://pure01.example.com/api/1.13/volume?filter=name='v1*'&sort=name-
```

Example 5: Get a list of either all volumes with names that include `123` and are greater than or equal to 10 gigabytes in size, or all volumes with names that include `abc` and are less than or equal to 10 gigabytes in size.

```
GET https://pure01.example.com/api/1.13/volume?filter=(contains(name,'123') and size >= 10737418240)
  or (contains(name,'abc') and size <= 10737418240)
```

Audit Trail

Every request that creates, modifies, or deletes a resource will be logged in the audit trail, which can be viewed through the Purity GUI (Messages > Audit Trail) and CLI (`puremessage`). For example, if the client sends the following request to create a snapshot of volume v1, the action, including the username of the user associated with the API token, is logged in the audit trail:

```
>> POST https://pure-001.example.com/api/1.13/volume

<< 200 OK

{
  "snap": true,
  "source": [
    "v1"
  ]
}
```

REST API and Scripting

Session Management

The client that makes the REST API requests will need to store a cookie returned by the REST service in order to continue to use the session. When using Python to create REST scripts, the CookieJar library can be used to manipulate session cookies. For example,

```
cj = CookieJar()
opener = urllib2.build_opener(urllib2.HTTPCookieProcessor(cj))
urllib2.install_opener(opener)
```

Refer to the cookielib module at docs.python.org for more information about cookie handling. Refer to RFC 2109 for protocol standards and recommendations about setting cookies.

Content Type

The request header must be set to the “application/json” content type. When using Python to create REST scripts, set the content-type header value to application/json. For example,

```
headers = {"Content-Type": "application/json"}
req = urllib2.Request(url, headers=headers)
```

Back to top

Release Notes

This section outlines the changes made to the REST API.

Compatibility Changes

The following compatibility changes affect all versions of the REST API and potentially break pre-existing API requests.

Description	Details	Purity Versions
Changes to **cert** requests	With support for named certificates, changed the meaning and behavior of the **GET cert** request. **GET cert** now returns all attributes of all certificates on the array. **cert** requests that work with previous releases of the API will need to be modified for use with REST API 1.12. (Pre-existing **cert** requests will continue to work with API 1.11 and earlier versions.)	4.10.7 and later
Changed the output of the **GET drive** and **GET drive/{drive}** requests	The output of the **GET drive** and **GET drive/{drive}** requests is changed when the value of the `type` field is null. The output previously returned "`-` in the `type` field for a null value. Now `null` is returned in the `type` field for a null value. This change affects all versions of the REST API, when connecting with FlashArrays running Purity 4.10.0 or later versions.	4.10.0 and later
Made **GET message** `open` and `audit` options incompatible	The `open` and `audit` parameters are not supported together in the same **GET message** REST API request. This change affects all versions of the REST API, when connecting with FlashArrays running Purity 4.5.0 or later versions.	4.5.0 and later
Deprecated DRV component name	The DRV component name is deprecated. This change affects the following requests: • Deprecated DRV information in the response of **GET hardware** requests. Information for the BAY component name is still reported. • Deprecated DRV in the **GET hardware/{component}** request URL. Instead, use the BAY component name with a **GET drive/{component}** request. For example, deprecated `GET https://pure01.example.com/api/1.4/hardware/SH0.DRV0.` Instead, use `GET https://pure01.example.com/api/1.4/drive/SH0.BAY0.` (Note: The BAY component name is still available with the **GET hardware/{component}** request, but the **GET drive/{component}** request above is preferred.) The DRV deprecation affects all versions of the REST API, when connecting with FlashArrays running Purity 4.1.4 or later versions.	4.1.4 and later
Removed method stub **PUT port**	Removed the method stub **PUT port**, which was inadvertently exposed. This removal affects all versions of the REST API, when connecting with FlashArrays running Purity 4.1.0 or later versions.	4.1.0 and later

TLS Requirements

With the release of Purity 4.7.0, support for TLS 1.0 is dropped for all versions of the REST API.

The following table lists the TLS versions required to communicate with the Pure Storage REST API:

Purity Version	Supported TLS Versions
Purity 4.7.0 and later	TLS 1.1 and 1.2
Earlier versions of Purity	TLS 1.0, 1.1, and 1.2

REST API 1.13

This section describes changes made between REST API 1.12 and REST API 1.13.

Description	Details
Added support for expiring API tokens	Added the `timeout` parameter to the **POST admin/{name}/apitoken** endpoint to cause the API token to expire after the provided number of seconds. Added the `expires` field to the output of the following requests. The `expires` field displays the time at which the token expires. • **GET admin** • **GET admin/{name}** • **GET admin/{name}/apitoken** • **POST admin/{name}/apitoken** • **DELETE admin/{name}/apitoken**
Added specification information to the output of hardware requests	The `model` and `serial` fields to the output of the following requests: • **GET hardware** • **GET hardware/{name}** • **PUT hardware/{name}**
Added support for getting TCP replication connections	The following endpoint returns all the TCP connections and their statuses: **GET array/connection/path**. Note that the TCP connections are only visible on the array that received the connection request.
Added support for ActiveCluster connections	Added `sync-replication` as a valid `type` in the **POST array/connection** request. Added `sync-replication` as a valid `type` in the response of **GET array/connection** requests.
Added support for volume groups	Added the following endpoints to support creating and managing volume groups: • **GET vgroup** to list the volume groups on an array. • **GET vgroup/{name}** to list information about a specific volume group. • **POST vgroup/{name}** to create a new volume group. • **PUT vgroup/{name}** to rename or modify a volume group or to recover a destroyed volume group. • **DELETE vgroup/{name}** to destroy a volume group.
Added support for virtual volumes	Added the `protocol_endpoint` parameter to the following endpoints. (Response output does not include the `size` field when this parameter is present.) • **GET volume** • **GET volume/{name}** • **POST volume/{name}** Added the `use_protocol_endpoint` parameter to the following endpoints. `use_protocol_endpoint` should refer to the name of a protocol_endpoint volume that is already connected to the host group or host. • **POST hgroup/{name}/volume/{volume}** • **POST host/{name}/volume/{volume}**
Added support for listing remote hosts, host groups, and connections	Added a `remote` parameter to several endpoints to also list information on remote arrays that are connected for `sync-replication`. With the `remote` parameter, the following requests also list hosts on remote arrays that are connected for `sync-replication`, as well as local hosts: • **GET host**

	GET host/{name} With the `remote` parameter, the following request also lists connections to known volumes on remote arrays that are connected for `sync-replication`, as well as connections to local hosts: • **GET host/{name}/volume** With the `remote` parameter, the following requests also list connections to known host groups on remote arrays that are connected for `sync-replication`, as well as local host groups: • **GET hgroup** • **GET hgroup/{name}** • **GET hgroup/{name}/volume** With the `remote` parameter, the following requests also list connections to known volumes on remote arrays that are connected for `sync-replication`, as well as local connections: • **GET volume** • **GET volume/{name}/host** • **PUT volume/{name}/hgroup**
Added support for viewing mirrored and per-array performance	Added these options to the following endpoints: • `mirrored`: to view performance statistics for I/Os that have been mirrored as part of an ActiveCluster connection • `array`: to view performance statistics for I/Os split by the array that the I/O originated on The affected endpoints are: • **GET volume** • **GET volume/volume}** • **GET host** • **GET host/{host}** • **GET hgroup** • **GET hgroup/{hgroup}** • **GET pod** • **GET pod/{pod}** Also added the `mirrored` option to the following endpoint: • **GET array** The `mirrored` and `array` parameters are optional and are only valid if `action=monitor`.

REST API 1.12

This section describes changes made between REST API 1.11 and REST API 1.12.

Description	Details
Added support for multiple certificates	Added two new endpoints: • **POST cert/{name}** to create a certificate with the specified name • **DELETE cert/{name}** to delete the specified certificate **cert** requests now take a `name` component: • **GET cert/{name}** to list the specified certificate • **PUT cert/{name}** to modify the specified certificate Added a `name` field to responses for **GET cert**, **GET cert/{name}**, and **PUT cert/{name}** requests. Added a `ca-certificate` field to responses for **GET cert/{name}**, **POST cert/{name}**, and **PUT cert/{name}** requests. **Notes**: • The management certificate, used internally by the Purity GUI, cannot be modified or deleted. • The **GET cert** request now returns all attributes of all certificates on the array.
Increased support for SNMP v3	Added `v3` as a supported value for the `version` parameter with the special `localhost` endpoint (which is associated with the array's SNMP agent).

	Added a `notification` parameter to the following endpoints. Supported values are `inform` and `trap`. • **POST snmp/{name}** • **PUT snmp/{name}** Added a `notification` field to responses for the following endpoints: • **GET snmp** • **GET snmp/{name}** • **POST snmp/{name}** • **PUT snmp/{name}** **Note**: In environments that include arrays runnings other versions of the REST API, we recommend using only REST 1.12 (or higher) to configure SNMP v3.
Added support for KMIP	Added support for the Key Management Interoperability Protocol (KMIP) with a Vormetric DSM server.
Added support for pod management	Added the following endpoints to create and manage pods: • **GET pod** to list or monitor pods. • **GET pod/{name}** to list information about or monitor a specfic pod. • **POST pod/{name}** to create a new pod. • **DELETE pod/{name}** to destroy a pod. • **PUT pod/{name}** to rename a pod or to recover a destroyed pod. • **POST pod/{name}/array/{array name}** to stretch a pod into an array. • **DELETE pod/{name}/array/{array name}** to unstretch a pod from an array.

REST API 1.11

This section describes changes made between REST API 1.10 and REST API 1.11.

Description	Details
Added new field to the output of **GET drive** and **GET drive/{drive}** requests	Added the `protocol` field to the output of **GET drive** and **GET drive/{drive}** requests. The `protocol` field currently returns these values: • `SAS`: for drives with SAS connections • `NVMe`: for drives with NVMe connections

REST API 1.10

This section describes changes made between REST API 1.9 and REST API 1.10.

Description	Details
Added SAN latency information	Changed the response of the **GET array?action=monitor**, **GET hgroup?action=monitor**, **GET host?action=monitor**, and **GET volume?action=monitor** requests. The following fields no longer include SAN latency: • `usec_per_read_op`: displays the average arrival-to-completion time for a host read operation (microseconds) • `usec_per_write_op`: displays the average arrival-to-completion time for a host write operation (microseconds) Added the following fields to report SAN latency: • `san_usec_per_read_op`: displays the average time required to transfer data from the array to the initiator (microseconds) • `san_usec_per_write_op`: displays the average time required to transfer data from the initiator to the array (microseconds)

REST API 1.9

This section describes changes made between REST API 1.8 and REST API 1.9.

Description	Details
Added support for Pure Apps	Added following endpoints to display information about installed apps:

- **GET app** to display information about all installed apps
- **GET app/{app}** to display information about a specific app

Added support for IPv6	On arrays running Purity 4.9.0 and above, IPv6 addresses can be passed anywhere an IPv4 address could be passed using any REST version. **Note**: In a mixed environment with some arrays running Purity 4.8.x or below, using REST API 1.9 guarantees that the request supports IPv6.

REST API 1.8

This section describes changes made between REST API 1.7 and REST API 1.8.

Compatibility Changes in REST API 1.8

Description	Details
Changed the response returned by the **GET drive** requests.	Removed the `degraded` field from the responses of the **GET drive** and **GET drive/{drive}** requests. Previously, the `degraded` field displayed the percentage of data that is not protected by redundancy. **Note**: This change affects all versions of the REST API. The `parity` field in the response of **GET array?space=true** request now provides data redundancy information that previously appeared in the `degraded` field. **Note**: The `parity` field is only available in version 1.8 and higher of the REST API.

Other REST API 1.8 Changes

Description	Details
Added information about data redundancy to the **GET array** response	Added the `parity` field, showing the percentage of data that is protected by RAID-6, to the response returned by the **GET array?space=true** request. The percentage value drops below 100% if the data is not fully protected, such as when a module is pulled and the array is rebuilding the data to bring it back to full parity. **Note**: The `parity` field returns `null` when the `historical` parameter is used.
Added voltage information to **GET hardware** responses	Added the `voltage` field to the responses of the **GET hardware** and **GET hardware/{component}** requests.

REST API 1.7

This section describes changes made between REST API 1.6 and REST API 1.7.

Compatibility Changes in REST API 1.7

Description	Details
Changed the response returned by protection group snapshot transfer	Removed the `queued` field from the response of the **GET pgroup/{pgroup}?snap=true&transfer=true** request. The `started` field now provides the data that previously appeared in the `queued` field. **Note**: This change also affects REST API 1.5 and REST API 1.6.
Changed the response returned by phonehome	In 4.7.x release and 4.6.x releases after 4.6.4, the response to the **GET array/phonehome** request was: `{"status": "", "action": ""}`. In 4.8.0, the response is changed back to `{"status": null, "action": null}` to be backwards compatible with 4.1.x and 4.5.x releases.

Other REST API 1.7 Changes

Description	Details
Added the ability to add and	Added the following endpoints to add or remove hosts, host groups, and volumes to and from protection groups:

remove hosts, host groups, and volumes from protection groups	• **POST host/{host}/pgroup/{pgroup}** to add a host to a protection group • **POST hgroup/{hgroup}/pgroup/{pgroup}** to add a host group to a protection group • **POST volume/{volume}/pgroup/{pgroup}** to add a volume to a protection group • **DELETE host/{host}/pgroup/{pgroup}** to remove a host from a protection group • **DELETE hgroup/{hgroup}/pgroup/{pgroup}** to remove a host group from a protection group • **DELETE volume/{volume}/pgroup/{pgroup}** to remove a volume from a protection group
Added the ability to list protection group membership for hosts, host groups, and volumes	Added the `protect` parameter to the following endpoints: • **GET host?protect=true** to display all protected hosts and their associated protection groups • **GET hgroup?protect=true** to display all protected host groups and their associated protection groups • **GET volume?protect=true** to display all protected volumes and their associated protection groups • **GET host/{host}?protect=true** to display protection group membership for a specific host • **GET hgroup/{hgroup}?protect=true** to display protection group membership for a specific host group • **GET volume/{volume}?protect=true** to display protection group membership for a specific volume Set the `protect` value to `true` to list protection group membership for hosts, host groups, and volumes. The `protect` parameter is optional.
Added support for filtering, limiting, pagination, and sorting	Filtering narrows down the results of a GET request. Limiting restricts the number of entries returned in the results of a GET request. Pagination displays the results of a GET request as page-by-page lists. Sorting returns the response objects of a GET request in the order specified. See the Sorting, Filtering, and Pagination section.
Added port failover status	The **GET port** endpoint now also returns the failover status of ports (NPIV status of FC ports).
Added ability to display private and shared connections and to display LUNs used in host-volume connections	Added the `connect` parameter to the following endpoints: • **GET hgroup** • **GET host** • **GET volume** Set the `connect` value to `true` to display private or shared host-volume connections and the LUNs used by the hosts to address the volumes. The `connect` parameter is optional. Added the `private` and `shared` parameters to the following endpoints: • **GET host** • **GET volume** Set the `private` value to `true` to display private host-volume connections and the LUNs used by the hosts to address the volumes. Set the `shared` value to `true` to display shared host-volume connections and the LUNs used by the hosts to address the volumes. The `private` and `shared` parameters are optional and are only valid if the `connect` parameter is set to `true`.
Added ability to query specific object names (for hosts, host groups, protection groups, and volumes)	Added the `names` parameter to the following endpoints to restrict the returned output to only the named objects: • **GET hgroup** • **GET host** • **GET pgroup** • **GET volume** The `names` parameter accepts a single name or a comma-separated list for multiple values. The `names` parameter also accepts an asterisk (*) as a wildcard character. The names parameter is optional, as is the asterisk wildcard character.
Added limited support for asterisk as a wildcard character	An asterisk (*) wildcard character representing zero or more unspecified characters is supported in filters and in the `names` parameter when used with the following objects: • Hosts • Host groups • Protection groups • Protection group snapshots • Volumes

	The asterisk wildcard character is optional.
Added size and total options to monitor resources	Added `size` parameter to the following endpoints: • **GET array** • **GET host** • **GET host/{host}** • **GET hgroup** • **GET hgroup/{hgroup}** • **GET volume** • **GET volume/{volume}** Set the `size` value to `true` to display the average I/O sizes. The `size` parameter is optional and is only valid if the `action` parameter is set to `monitor`. Added the `total` parameter to the following endpoints: • **GET hgroup** • **GET host** • **GET volume** Set the `total` value to `true` to append a row containing aggregated data. The `total` parameter is optional and is only valid if the `action` parameter is set to `monitor`.

REST API 1.6

The following changes were made between REST API 1.5 and REST API 1.6:

Description	Details
Added ability to list historical storage capacity and utilization data	For the **GET array**, **GET volume**, and **GET volume/{volume}** requests, added the ability to use the `historical` parameter with the `space=true` parameter to display historical storage capacity and utilization data.

Note: See also REST 1.7 Compatibility Changes for a change to the response returned by a protection group snapshot transfer request. This change is also made in REST 1.5 and REST 1.6.

REST API 1.5

The following changes were made between REST API 1.4 and REST API 1.5:

Description	Details
Added ability to list performance data for hosts, host groups, volumes, or snapshots	Added the `action=monitor` value to the following requests to display real-time I/O performance data: • **GET host** to display data for all hosts • **GET host/{host}** to display data for the specified host • **GET hgroup** to display data for all hosts • **GET hgroup/{hgroup}** to display data for the specified host Added the `action=monitor` value to the **GET volume** request to display real-time or historical performance data. Combine with the `historical` parameter to display historical data. The `action` parameter is optional.
Added ability to specify multiple protection groups with the `pgrouplist` parameter	The **GET volume** and **GET volume/{volume}** requests now accept either a single protection group or a comma-separated list of protection groups in the `pgrouplist` parameter. When multiple protection groups are specified, the request returns information for all snapshots that were created for any of the specified protection groups. Previously, the `pgrouplist` parameter accepted only a single protection group.
Added ability to specify optional Boolean parameters as `false`	This code change does not modify API request behavior, except to allow an optional Boolean parameter to be set to `false`. Setting an optional Boolean parameter to `false` is equivalent to not including the parameter in the request. Previously, optional Boolean parameters only accepted `true`. The following are examples of optional Boolean parameters that now also accept `false`: • GET admin?api_token • GET admin?expose • GET admin?publickey

	• GET array?controllers • GET host?all
Added ability to copy protection groups	Added the `source` and `overwrite` parameters to the **POST pgroup/{pgroup}** request to enable the copying of protection groups. The `source` and `overwrite` parameters are optional.
Added ability to manage subnets and virtual network interfaces	Added the following resources to support VLAN tagging: • **GET subnet** and **GET subnet/{subnet}** to list subnets • **POST subnet/{subnet}** to create a subnet • **PUT subnet/{subnet}** to set subnet attributes • **DELETE subnet/{subnet}** to delete a subnet • **POST network/vif** to create a VLAN interface • **DELETE network/vif** to delete a VLAN interface
Added ability to manage traffic between connected arrays	Added the **PUT array/connection/{connection}** resource to modify bandwidth throttling attributes for a connected array.
Added controller name to **GET array/remoteassist** response	Changed **GET array/remoteassist** to include the controller's hostname in the request response.
Removed ability to expose the API tokens of other users	Changed the **GET admin** `expose` parameter to unmask only the API token of the current user when set to `true`. Previously, setting the `expose` parameter to `true` unmasked the API tokens of all users.

Note: See also REST 1.7 Compatibility Changes for a change to the response returned by a protection group snapshot transfer request. This change is also made in REST 1.5 and REST 1.6.

REST API 1.4

The following changes were made between REST API 1.3 and REST API 1.4:

Description	Details
Added ability to list user login sessions	Added the **GET message** `login` parameter to list user session logs. Include the recent parameter with `login` to display login, logout, and authentication events that have occurred within the past 24 hours. Include the user parameter with `login` to list user session logs for a specific user.
Added ability to list volume snapshots created as part of one or more protection groups or protection group snapshots	When the **GET volume** `pgrouplist` parameter is set, the REST API lists all volume snapshots that were created as part of protection group snapshots for the specified protection group or protection group snapshot. When the **GET volume/{volume}** `pgrouplist` parameter is set, the REST API lists the volume snapshots of the specified volume that were created as part of protection group snapshots for the specified protection group. The `pgrouplist` parameter is optional and is only valid if `snap` is true.
Added ability to recover and rename volume snapshots	Added the ability to recover and rename volume snapshots through **PUT volume/{volume}**. Previously, **PUT volume{volume}** resized, recovered and renamed volume names only. Now, **PUT volume{volume}** can also recover and rename volume snapshots. When renaming a volume snapshot, only the suffix can be changed.
Added ability to generate a snapshot for multiple protection groups specified	Added the ability to generate a snapshot for multiple protection groups specified through the **POST pgroup** `source` parameter. Previously, the source parameter only accepted one protection group. Now, the source parameter accepts a list of protection groups. The source parameter is required and is only valid if `snap` is true.
Added ability to specify whether snapshot data should be immediately replicated, or queued up to begin replicating after all earlier replication sessions are complete	Added the ability to specify whether snapshot data should be: • Immediately replicated (**POST pgroup** `replicate_now`) to each allowed target, or • Queued up (**POST pgroup** `replicate`) and only begin replicating to each allowed target after all earlier replication sessions for the same protection group have been completed to that target.

	When the **POST pgroup** `replicate_now` parameter is set, the REST API immediately replicates the snapshot data to all allowed target arrays. When the **POST pgroup** `replicate` parameter is set, the REST API queues the snapshot for replication to all allowed target arrays, excluding sessions started with `replicate_now`. Purity will begin replicating data to each allowed target when all earlier replication sessions for the same protection group have been completed to that target. The `replicate_now` and `replicate` parameters are optional and cannot be used together.
Added ability to specify a preferred time of day at which to generate local snapshots	Added the ability to specify the preferred time, on the hour, at which to generate local snapshots through the **PUT pgroup/{pgroup}** `snap_at` parameter. The `snap_at` parameter is optional and is only valid if `snap_frequency` is set to 3600 or more seconds.
Changed **POST snmp/{manager}** `host` parameter to be a required parameter	Changed the **POST snmp/{manager}** `host` parameter from an optional to required parameter.
Deprecated ability to list all volume snapshots that have been created as part of protection group snapshots	Deprecated the ability to list all volume snapshots that have been created as part of protection group snapshots through the **GET volume** `pgroup` and **GET volume/{volume}** `pgroup` parameters. A new parameter named `pgrouplist` has been created to perform similar requests.

REST API 1.3

The following changes were made between REST API 1.2 and REST API 1.3:

Description	Details
Added ability to view and manage SSL certificates	Added the following resources to view and manage SSL certificates: • **GET cert** to list certificate attributes or export certificates. • **GET cert/certificate_signing_request** to construct a certificate signing request (CSR) for signing by a certificate authority (CA). • **PUT cert** to create a self-signed certificate or import a certificate signed by a certificate authority (CA).
Added ability to clear the `replicate_at` time	Added the ability to clear the **PUT pgroup/{pgroup}** `replicate_at` parameter by passing in a null value.
Added ability to clear the `replicate_blackout` period	Added the ability to clear the **PUT pgroup/{pgroup}** `replicate_blackout` parameter by passing in a null value.
Changed the response of the `replicate_blackout` field	If the **PUT pgroup/{pgroup}** `replicate_blackout` parameter is set to a null value, the `replicate_blackout` field for **GET pgroup/{pgroup}** returns an empty response. Before, `replicate_blackout` returned the response `[{"start": 0,"end": 0}]`.

REST API 1.2

The following changes were made between REST API 1.1 and REST API 1.2:

Description	Details
Added ability to connect and disconnect arrays	Added the following resources to manage array connections: • **GET array/connection** to list the connected arrays. • **GET array?connection_key=true** to list the connection key. • **POST array/connection** to connect an array. • **DELETE array/connection/{connection}** to disconnect an array.
Added ability to manage protection groups	Added the following resources to manage protection groups: • **GET pgroup** to list protection groups and their attributes. • **GET pgroup/{pgroup}** to list the attributes for the specified protection group. • **POST pgroup{pgroup}** to create a protection group with the specified name. • **PUT pgroup{pgroup}** to modify a protection group, its retention schedule, and its snapshot schedule. • **DELETE pgroup{pgroup}** to destroy a protection group and all of its snapshots.

Deprecated ability to rename volume snapshots	Deprecated the ability to rename snapshots through the **PUT volume/{volume}** `name` parameter. If the Purity client attempts to rename snapshots, the REST API returns response error code `400 BAD REQUEST`. This change also impacts REST API versions 1.0 and 1.1.
Added various fields to the **POST volume/{volume}** response	Added the following fields to the **POST volume/{volume}** response: `source`, `serial`, `size` and `created`.

REST API 1.1

The following changes were made between REST API 1.0 and REST API 1.1:

Description	Details
Added ability to overwrite an existing volume	Added the `overwrite` parameter to **POST volume/{volume}**. When used with `source`, set the `overwrite` value to `true` to overwrite an existing volume. Boolean (true or false). Optional.
Renamed parameter from force to trust	Renamed the **PUT directoryservice** parameter from `force` to `trust`. When used with certificate, set the `trust` value to `true` to skip certificate chain trust verification. Boolean (true or false). Optional.
Fixed a bug in **GET volume**	In v1.0, calls to **GET volume** with `space` set to `true` failed to return values for `thin_provisioning` and `total_reduction`. This has been fixed.

Back to top

Resources

1. Authentication

Create and destroy REST sessions.

1.1 POST auth/apitoken

Generates a REST API token that can be used to create a REST session.

API tokens are tied to a particular administrative account. Each administrator has permission to manage their own API token. Administrators cannot manage

tokens on behalf of other users.

Parameters

Parameter	Type	Description
password	string	Purity user login password used to generate the REST API token. Required.
username	string	Purity user login name used to generate the REST API token. Sometimes referred to as sAMAccountName. Required.

Example 1: Create an API token

Request:

```
POST https://pure01.example.com/api/1.13/auth/apitoken
{
  "password": "pureuser",
  "username": "pureuser"
}
```

Response:

```
{
  "api_token": "d28f726a-1d62-9850-644d-1f08b59d6141"
}
```

1.2 POST auth/session

Creates a REST session using the REST API token.

Parameters

Parameter	Type	Description
api_token	string	REST API token used to create a REST session. Required.

Example 1: Create a REST session

Request:

```
POST https://pure01.example.com/api/1.13/auth/session
{
  "api_token": "$APITOKEN"
}
```

Response:

```
{
  "username": "os76"
}
```

1.3 DELETE auth/session

Deletes (invalidates) the REST session.

Parameters

Parameter	Type	Description
None	n/a	n/a

Example 1: Delete (invalidate) a REST session

Request:

```
DELETE https://pure01.example.com/api/1.13/auth/session
```

Response:

```
{
  "username": "os76"
}
```

2. Array

View and manage the array, and administer the phonehome and remote assist features.

2.1 GET array

Lists the attributes for the array, including the array name, Purity version and Purity revision number. Can also display historical performance or capacity data.

Parameters

Parameter	Type	Description
action	string	If set to `monitor`, displays the following real-time or historical performance data: **Latency** • `san_usec_per_read_op` - average time, measured in microseconds, required to transfer data from the array to the initiator. • `san_usec_per_write_op` - average time, measured in microseconds, required to transfer data from the initiator to the array. • `usec_per_read_op` - average time, measured in microseconds, it takes the array to process an I/O read request. • `usec_per_write_op` - average time, measured in microseconds, it takes the array to process an I/O write request. • `queue_depth` - average number of queued I/O requests. **IOPS** • `reads_per_sec` - number of read requests processed per second. • `writes_per_sec` - number of write requests processed per second. **Bandwidth** • `input_per_sec` - number of bytes written per second. • `output_per_sec` - number of bytes read per second. Optional.
banner	boolean	Displays the "message of the day" (MOTD) text that is sent to all Purity users. The banner message is displayed in the login pane of the Purity GUI and via SSH after users log in. Optional.

connection_key	boolean	Displays a connection key that can be used to connect to this array. Optional.
controllers	boolean	Displays the mode and status of each controller on the array. For arrays to which a second controller has never been connected, only a single controller mode is displayed. Once an additional controller has been connected, its information is always shown. Optional.
historical	string	If the `action` parameter is set to `monitor`, set this parameter to display historical performance data at the specified resolution. If the `space` parameter is set to `true`, set this parameter to display historical space data at the specified resolution. Valid `historical` values are: `1h`, `3h`, `24h`, `7d`, `30d`, `90d`, and `1y`. Optional.
idle_timeout	boolean	Displays the idle time limit, in minutes, of the Purity GUI and CLI sessions. Users are automatically logged out a session that exceeds the idle timeout limit. Optional.
ntpserver	boolean	Displays the hostnames or IP addresses of the NTP servers currently being used by the array to maintain reference time. Optional.
phonehome	boolean	Displays the current state of the Purity phonehome automatic hourly log transmission facility as enabled or disabled. Optional.
proxy	boolean	Displays the proxy host for the phonehome facility when HTTPS is the phonehome protocol (the phonehome facility itself determines which protocol to use). Optional.
relayhost	boolean	Displays the hostname or IP address of the electronic mail relay server currently being used as a forwarding point for email alerts generated by the array. Optional.
scsi_timeout	boolean	Displays the amount of time (in seconds) that can lapse during an I/O interruption before the target ports log out of the fabric. The default timeout value is 60 seconds. Optional.
senderdomain	boolean	Displays domain name from which Purity sends email alert messages. Optional.
space	boolean	Displays the amount of usable physical storage on the array and the amount of storage occupied by data and metadata. Optional.
syslogserver	boolean	Displays remote syslog servers for delivering notifications. Optional.

Example 1: List array attributes

Request:

```
GET https://pure01.example.com/api/1.13/array
```

Response:

```
{
  "array_name": "pure01",
  "id": "c35e0c1f-5ce9-4143-9121-e78bd57619a6",
  "revision": "201710050655+ef2a03a",
  "version": "5.0.0"
}
```

Example 2: List array controllers

Request:

```
GET https://pure01.example.com/api/1.13/array?controllers=true
```

Response:

```
[
  {
    "mode": "primary",
    "model": "VMware",
    "name": "CT0",
    "status": "ready",
    "version": "5.0.0"
  }
]
```

Example 3: List array space usage

Request:

```
GET https://pure01.example.com/api/1.13/array?space=true
```

Response:

```
[
  {
    "capacity": 3510211154,
    "data_reduction": 1.0,
    "hostname": "pure01",
    "parity": 1.0,
    "shared_space": 0,
    "snapshots": 0,
    "system": 0,
    "thin_provisioning": 0.0,
    "total": 0,
    "total_reduction": 1.0,
    "volumes": 0
  }
]
```

Example 4: List array space usage for the past hour

Request:

```
GET https://pure01.example.com/api/1.13/array?space=true&historical=1h
```

Response:

```
[
  {
    "capacity": 3518741436,
    "data_reduction": 1.0,
    "hostname": "pure01",
    "parity": null,
    "shared_space": 0,
    "snapshots": 0,
    "system": 0,
    "thin_provisioning": null,
    "time": "2017-10-06T21:56:47Z",
    "total": 0,
    "total_reduction": null,
```

```
    "volumes": 0
  }
]
```

Example 5: List real-time performance data for the array

Request:

```
GET https://pure01.example.com/api/1.13/array?action=monitor
```

Response:

```
[
  {
    "input_per_sec": 0,
    "output_per_sec": 0,
    "queue_depth": 0,
    "reads_per_sec": 0,
    "san_usec_per_read_op": 0,
    "san_usec_per_write_op": 0,
    "time": "2017-10-06T22:00:29Z",
    "usec_per_read_op": 0,
    "usec_per_write_op": 0,
    "writes_per_sec": 0
  }
]
```

Example 6: List historical performance data for the array at resolution '1h'

Request:

```
GET https://pure01.example.com/api/1.13/array?action=monitor&historical=1h
```

Response:

```
[
  {
    "input_per_sec": 0,
    "output_per_sec": 0,
    "queue_depth": 0,
    "reads_per_sec": 0,
    "san_usec_per_read_op": 0,
    "san_usec_per_write_op": 0,
    "time": "2017-10-06T21:56:47Z",
    "usec_per_read_op": 0,
    "usec_per_write_op": 0,
    "writes_per_sec": 0
  },
  {
    "input_per_sec": 0,
    "output_per_sec": 0,
    "queue_depth": 0,
    "reads_per_sec": 0,
    "san_usec_per_read_op": 0,
    "san_usec_per_write_op": 0,
    "time": "2017-10-06T21:57:17Z",
    "usec_per_read_op": 0,
    "usec_per_write_op": 0,
    "writes_per_sec": 0
  },
...
    "usec_per_write_op": 0,
    "writes_per_sec": 0
  }
]
```

2.2 GET array/connection

Lists connected arrays.

Parameters

Parameter	Type	Description

throttle	boolean	Displays (`true`) outgoing bandwidth throttling information. Optional.

Example 1: List connected arrays

Request:

```
GET https://pure01.example.com/api/1.13/array/connection
```

Response:

```
[]
```

2.3 GET array/console_lock

Displays the status of the console lock.

Parameters

Parameter	Type	Description
None	n/a	n/a

Example 1: Displays the console lock status

Request:

```
GET https://pure01.example.com/api/1.13/array/console_lock
```

Response:

```
{
  "console_lock": "disabled"
}
```

2.4 GET array/phonehome

Lists information about the status of the transmission logs for the phonehome facility.

Parameters

Parameter	Type	Description
None	n/a	n/a

Example 1: Display phonehome status

Request:

```
GET https://pure01.example.com/api/1.13/array/phonehome
```

Response:

```
{
  "action": null,
  "status": null
}
```

2.5 GET array/remoteassist

Lists information about the status (enabled or disabled) of a remote assist session.

Parameters

Parameter	Type	Description
None	n/a	n/a

Example 1: Display remoteassist status

Request:

```
GET https://pure01.example.com/api/1.13/array/remoteassist
```

Response:

```
{
  "name": "pure01-ct0",
  "port": "",
  "status": "disabled"
}
```

2.6 POST array/connection

Creates a connection between the current array and the specified array.

Parameters

Parameter	Type	Description
connection_key	string	The connection_key of the array to be connected. Required.
management_address	string	The management address of the array to be connected. Required.
replication_address	string	The replication address of the array to be connected. Optional.
type	list	The type(s) of connection desired. The only option supported in this version is '`replication`'. Required.

Example 1: Create a connection between array pure01 and array pure02 for replication purposes.

Request:

```
POST https://pure01.example.com/api/1.13/array/connection
{
  "connection_key": "$CONNECTION_KEY",
  "management_address": "pure02.example.com",
  "type": [
    "replication"
  ]
}
```

Response:

```
{
  "array_name": "pure02",
```

```
  "connected": true,
  "id": "4821b86a-5d22-4361-a567-620e88bf78c8",
  "management_address": "pure02",
  "replication_address": [
    "10.14.225.209"
  ],
  "throttled": null,
  "type": [
    "async-replication"
  ],
  "version": "5.0.0"
}
```


2.7 PUT array

Modifies the attributes for the array.

At least one parameter must be specified.

Parameters

Parameter	Type	Description
banner	string	Sets a common "message of the day" (MOTD) that is sent to all Purity users. The banner message is displayed in the login pane of the Purity GUI and via SSH after users log in. Optional.
idle_timeout	number	Sets the idle time limit, in minutes, of the Purity GUI and CLI sessions. Valid values are between 5 and 180 minutes. The default timeout value is 30 minutes. Specifying a value of zero disables the automatic log-off feature. Changes made to the `idle_timeout` value do not apply to existing Purity sessions. Optional.
name	string	Renames the array. Optional.
ntpserver	list	Specifies alternate NTP servers, by IP address or hostname, assigned as the array source for reference time. Supersedes any previous NTP server assignments. If specifying an IP address, for IPv4, specify the IP address in the form `ddd.ddd.ddd.ddd`, where `ddd` is a number ranging from 0 to 255 representing a group of 8 bits. For IPv6, specify the IP address in the form `xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx`, where `xxxx` is a hexadecimal number representing a group of 16 bits. When specifying an IPv6 address, consecutive fields of zeros can be shortened by replacing the zeros with a double colon (`::`). Optional.
proxy	string	Sets the proxy host for the phonehome facility when HTTPS is the phonehome protocol (the phonehome facility itself determines which protocol to use). The format for the value is `https://HOSTNAME:PORT`, where HOSTNAME is the name of the proxy host and PORT is the TCP/IP port number used by the proxy host. Optional.
relayhost	string	Sets the hostname or IP address of the electronic mail relay server currently being used as a forwarding point for email alerts generated by the array. If specifying an IP address, enter the IPv4 or IPv6 address. For IPv4, specify the IP address in the form `ddd.ddd.ddd.ddd`, where ddd is a number ranging from 0 to 255 representing a group of 8 bits. If a port number is also specified, append it to the end of the address in the format `ddd.ddd.ddd.ddd:PORT`, where `PORT` represents the port number. For IPv6, specify the IP address in the form `xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx`, where `xxxx` is a hexadecimal number representing a group of 16 bits. Consecutive fields of zeros can be shortened by replacing the zeros with a double colon (`::`). If a port number is also specified, enclose the entire address in square brackets (`[]`) and append the port number to the end of the address. For example, `[xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx]:PORT`, where `PORT` represents the port number.

		Optional.
scsi_timeout	number	Changes the amount of time, in seconds, that can lapse during an I/O interruption before the target ports log out of the fabric. The default timeout value is 60 seconds. Changing the default timeout value may cause an initiator to mistakenly interpret the status of the FlashArray as failed or generate a host timeout. Contact the Pure Storage Support team before you change the `scsi_timeout` value. Optional.
senderdomain	string	Sets the domain name from which Purity sends email alert messages. Optional.
syslogserver	list	Specifies the remote syslog servers for delivering notifications. For example, `tcp://MyHost.com`. Specify the URI in the format `PROTOCOL://HOSTNAME:PORT`. `PROTOCOL` is "tcp", "tls", or "udp". `HOSTNAME` is the syslog server hostname or IP address. If specifying an IP address, for IPv4, specify the IP address in the form `ddd.ddd.ddd.ddd`, where `ddd` is a number ranging from 0 to 255 representing a group of 8 bits. For IPv6, specify the IP address in the form `[xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx]`, where `xxxx` is a hexadecimal number representing a group of 16 bits. Enclose the entire address in square brackets (`[]`). Consecutive fields of zeros can be shortened by replacing the zeros with a double colon (`::`). `PORT` is the port at which the server is listening. If a port number is specified, append it to the end of the address. If the port is not specified, it defaults to 514. Optional.

2.8 PUT array/connection/{connection}

Modifies the bandwidth throttling attributes for a connected array.

At least one parameter must be specified.

Parameters

Parameter	Type	Description
default_limit	number or string	Default maximum bandwidth threshold for outbound traffic. Once exceeded, bandwidth throttling occurs. Enter the size as a number (bytes) or as a string with a single character unit symbol. Valid unit symbols are `K`, `M`, `G`, representing KiB, MiB, GiB, respectively, where "Ki" denotes 2^10, "Mi" denotes 2^20, and so on. If the unit symbol is not specified, the unit defaults to bytes. Optional.
window	object	Range of time during which `--window_limit` threshold is in effect. The start and end times must be set on the hour. Enter the time range with the `start` and `end` attributes, both measured in seconds, where `start` represents the window start time and `end` represents the window end time. Optional.
window_limit	number or string	Maximum bandwidth threshold for outbound traffic during the specified `window_limit` time range. Once exceeded, bandwidth throttling occurs. Enter the size as a number (bytes) or as a string with a single character unit symbol. Valid unit symbols are `K`, `M`, `G`, representing KiB, MiB, GiB, respectively, where "Ki" denotes 2^10, "Mi" denotes 2^20, and so on. If the unit symbol is not specified, the unit defaults to bytes. Optional.

2.9 PUT array/console_lock

Enables or disables root login through the console.

At least one parameter must be specified.

Parameters

Parameter	Type	Description
enabled	boolean	Enables (`true`) or disables (`false`) the console lock which prevents the root user from logging in through the system console. Optional.

Example 1: Enable console lock

Request:

```
PUT https://pure01.example.com/api/1.13/array/console_lock
{
  "enabled": true
}
```

Response:

```
{
  "console_lock": "enabled"
}
```

Example 2: Disable console lock

Request:

```
PUT https://pure01.example.com/api/1.13/array/console_lock
{
  "enabled": false
}
```

Response:

```
{
  "console_lock": "disabled"
}
```

2.10 PUT array/phonehome

Performs phonehome actions.

At least one parameter must be specified.

Parameters

Parameter	Type	Description
action	string	Transmits event logs stored in the array to the Pure Storage Support team via the phonehome channel. Specify the phonehome log time period as any of the following: send_all, send_today, send_yesterday, cancel. Optional.
enabled	boolean	Enables (`true`) or disables (`false`) the automatic hourly transmission of array logs to the Pure Storage Support team. Optional.

Example 1: Enable phonehome

Request:

```
PUT https://pure01.example.com/api/1.13/array/phonehome
{
  "enabled": true
}
```

Response:

```
{
  "phonehome": "enabled"
}
```

Example 2: Disable phonehome

Request:

```
PUT https://pure01.example.com/api/1.13/array/phonehome
{
  "enabled": false
}
```

Response:

```
{
  "phonehome": "disabled"
}
```

2.11 PUT array/remoteassist

Enables or disables a remote assist session.

Parameters

Parameter	Type	Description
action	string	Enables (`"connect"`) or disables (`"disconnect"`) a remote assist session. Required.

2.12 DELETE array/connection/{connection}

Disconnects the current array from the specified array.

Parameters

Parameter	Type	Description
None	n/a	n/a

Example 1: Disconnect array pure02 from array pure01

Request:

```
DELETE https://pure01.example.com/api/1.13/array/connection/pure02
```

Response:

```
{
  "name": "pure02"
}
```

3. Volumes and Snapshots

View and manage volumes and snapshots.

3.1 GET volume

Lists volumes and snapshots and their attributes.

Created time is displayed in Greenwich Mean Time (GMT) and size is displayed in bytes.

Parameters

Parameter	Type	Description
action	string	If set to `monitor`, displays real-time or historical performance data. Time is displayed in Greenwich Mean Time (GMT). Optional.
connect	boolean	Displays all host-volume connections, either private or shared, and the LUNs used by the hosts to address the volumes. Optional.
historical	string	If the `action` parameter is set to `monitor`, set this parameter to display historical performance data at the specified resolution. If the `space` parameter is set to `true`, set this parameter to display historical space data at the specified resolution. Valid `historical` values are: `1h`, `3h`, `24h`, `7d`, `30d`, `90d`, and `1y`. Include the `names` parameter to only display historical performance or space data for the specified volumes. Optional.
names	list	Displays specified volumes. Enter multiple volume names in comma-separated format. Volume names can include asterisk (`*`) wildcard characters to represent zero or more characters. Optional.
pending	boolean	Includes (`true`) destroyed volumes that are in the eradication pending state. Time remaining is displayed in seconds. Optional.
pending_only	boolean	Lists (`true`) destroyed volumes that are in the eradication pending state. Time remaining is displayed in seconds. Optional.
pgrouplist	list	Displays specified protection groups or protection group snapshots. Only valid if the `snap` parameter is set to `true`. Enter multiple names in comma-separated format. Names can include asterisk (`*`) wildcard characters to represent zero or more characters. Optional.
private	boolean	Used with the `connect` parameter to display (`true`) private host-volume connections. Optional.
protect	boolean	Displays (`true`) all protected volumes and their associated protection groups. Optional.

shared	boolean	Used with the `connect` parameter to display (`true`) shared host-volume connections. Optional.
snap	boolean	Lists all snapshots (`true`). Optional.
space	boolean	Lists size and space consumption attributes for each volume. Optional.

Example 1: List all volumes

Request:

```
GET https://pure01.example.com/api/1.13/volume
```

Response:

```
[
  {
    "created": "2017-10-06T22:00:22Z",
    "name": "v1",
    "serial": "C35E0C1F5CE9414300011011",
    "size": 1073741824,
    "source": null
  },
  {
    "created": "2017-10-06T22:00:22Z",
    "name": "v2",
    "serial": "C35E0C1F5CE9414300011012",
    "size": 2147483648,
    "source": null
  },
  {
    "created": "2017-10-06T22:00:23Z",
    "name": "v3",
    "serial": "C35E0C1F5CE9414300011013",
    "size": 3221225472,
    "source": null
  }
]
```

Example 2: List all volumes including those pending eradication

Request:

```
GET https://pure01.example.com/api/1.13/volume?pending=true
```

Response:

```
[
  {
    "created": "2017-10-06T22:00:22Z",
    "name": "v1",
    "serial": "C35E0C1F5CE9414300011011",
    "size": 1073741824,
    "source": null,
    "time_remaining": null
  },
  {
    "created": "2017-10-06T22:00:22Z",
    "name": "v2",
    "serial": "C35E0C1F5CE9414300011012",
    "size": 2147483648,
    "source": null,
    "time_remaining": null
  },
  {
    "created": "2017-10-06T22:00:23Z",
    "name": "v3",
    "serial": "C35E0C1F5CE9414300011013",
    "size": 3221225472,
    "source": null,
    "time_remaining": null
  },
...
```

```
    "source": null,
    "time_remaining": 86400
  }
]
```

Example 3: List all volumes pending eradication

Request:

```
GET https://pure01.example.com/api/1.13/volume?pending_only=true
```

Response:

```
[
  {
    "created": "2017-10-06T22:00:23Z",
    "name": "v4",
    "serial": "C35E0C1F5CE9414300011014",
    "size": 4294967296,
    "source": null,
    "time_remaining": 86400
  }
]
```

Example 4: List all snapshots

Request:

```
GET https://pure01.example.com/api/1.13/volume?snap=true
```

Response:

```
[
  {
    "created": "2017-10-06T22:00:24Z",
    "name": "v3.1",
    "serial": "C35E0C1F5CE9414300011015",
    "size": 3221225472,
    "source": "v3"
  },
  {
    "created": "2017-10-06T22:00:24Z",
    "name": "v3.snap1",
    "serial": "C35E0C1F5CE9414300011016",
    "size": 3221225472,
    "source": "v3"
  },
  {
    "created": "2017-10-06T22:00:26Z",
    "name": "pg1.monday.v1",
    "serial": "C35E0C1F5CE9414300011017",
    "size": 1073741824,
    "source": "v1"
  },
  {
    "created": "2017-10-06T22:00:27Z",
    "name": "pg1.2.v1",
...
    "size": 1073741824,
    "source": "v1"
  }
]
```

Example 5: List per volume space usage

Request:

```
GET https://pure01.example.com/api/1.13/volume?space=true
```

Response:

```
[
  {
    "data_reduction": 1.0,
```

```
    "name": "v1",
    "shared_space": null,
    "size": 1073741824,
    "snapshots": 0,
    "system": null,
    "thin_provisioning": 1.0,
    "total": 0,
    "total_reduction": 1.0,
    "volumes": 0
  },
  {
    "data_reduction": 1.0,
    "name": "v2",
    "shared_space": null,
    "size": 2147483648,
    "snapshots": 0,
    "system": null,
    "thin_provisioning": 1.0,
    "total": 0,
    "total_reduction": 1.0,
    "volumes": 0
  },
...
    "total_reduction": 1.0,
    "volumes": 0
  }
]
```

Example 6: List all volume snapshots that were created as part of protection group snapshots for protection groups pg1 and pg2.

Request:

```
GET https://pure01.example.com/api/1.13/volume?snap=true&pgrouplist=pg1,pg2
```

Response:

```
[
  {
    "created": "2017-10-06T22:00:26Z",
    "name": "pg1.monday.v1",
    "serial": "C35E0C1F5CE9414300011017",
    "size": 1073741824,
    "source": "v1"
  },
  {
    "created": "2017-10-06T22:00:27Z",
    "name": "pg1.2.v1",
    "serial": "C35E0C1F5CE9414300011018",
    "size": 1073741824,
    "source": "v1"
  },
  {
    "created": "2017-10-06T22:00:27Z",
    "name": "pg2.1.v1",
    "serial": "C35E0C1F5CE9414300011019",
    "size": 1073741824,
    "source": "v1"
  }
]
```

3.2 GET volume/{volume}

Lists the attributes for the specified volume or snapshot.

If listing volume snapshot attributes, the `snap` parameter must be specified.

Parameters

Parameter	Type	Description
action	string	If set to `monitor`, displays real-time or historical performance data. Time is displayed in Greenwich Mean Time (GMT). Optional.
historical	string	If the `action` parameter is set to `monitor`, set this parameter to display historical performance data at the specified resolution.

		If the `space` parameter is set to `true`, set this parameter to display historical space data at the specified resolution. Valid `historical` values are: `1h`, `3h`, `24h`, `7d`, `30d`, `90d`, and `1y`. Optional.
pending	boolean	If the volume is in the eradication pending state, set pending to `true` to display the attributes for the destroyed volume. Optional.
pending_only	boolean	Lists (`true`) destroyed volumes that are in the eradication pending state and their attributes. Optional.
pgrouplist	list	Displays specified protection groups or protection group snapshots. Only valid if `snap` is true. Only valid if the `snap` parameter is set to `true`. Enter multiple names in comma-separated format. Names can include asterisk (`*`) wildcard characters to represent zero or more characters. Optional.
protect	boolean	Displays (`true`) the names of all protection groups associated with the specified volume. If the volume is not protected, returns an empty response. Optional.
snap	boolean	If listing the attributes for a specified volume, set `snap` to `true` to list all snapshots within the volume. If listing the attributes for a specified volume snapshot, always set `snap` to `true`. Optional if listing attributes for a specified volume. Required if listing attributes for a specified volume snapshot.
space	boolean	Lists size and space consumption attributes for the specified volume. Optional.

Example 1: List details for volume v1

Request:

```
GET https://pure01.example.com/api/1.13/volume/v1
```

Response:

```
{
  "created": "2017-10-06T22:00:22Z",
  "name": "v1",
  "serial": "C35E0C1F5CE9414300011011",
  "size": 1073741824,
  "source": null
}
```

Example 2: List snapshots of volume v3

Request:

```
GET https://pure01.example.com/api/1.13/volume/v3?snap=true
```

Response:

```
[
  {
    "created": "2017-10-06T22:00:24Z",
    "name": "v3.1",
    "serial": "C35E0C1F5CE9414300011015",
    "size": 3221225472,
    "source": "v3"
  },
  {
    "created": "2017-10-06T22:00:24Z",
```

```
    "name": "v3.snap1",
    "serial": "C35E0C1F5CE9414300011016",
    "size": 3221225472,
    "source": "v3"
  }
]
```

Example 3: List real-time performance data for volume v3.

Request:

```
GET https://pure01.example.com/api/1.13/volume/v3?action=monitor
```

Response:

```
[
  {
    "input_per_sec": 0,
    "name": "v3",
    "output_per_sec": 0,
    "reads_per_sec": 0,
    "san_usec_per_read_op": 0,
    "san_usec_per_write_op": 0,
    "time": "2017-10-06T22:00:32Z",
    "usec_per_read_op": 0,
    "usec_per_write_op": 0,
    "writes_per_sec": 0
  }
]
```

Example 4: List volume snapshots of v1 that were created as part of protection group snapshots for protection groups pg1 and pg2.

Request:

```
GET https://pure01.example.com/api/1.13/volume/v1?snap=true&pgrouplist=pg1,pg2
```

Response:

```
[
  {
    "created": "2017-10-06T22:00:26Z",
    "name": "pg1.monday.v1",
    "serial": "C35E0C1F5CE9414300011017",
    "size": 1073741824,
    "source": "v1"
  },
  {
    "created": "2017-10-06T22:00:27Z",
    "name": "pg1.2.v1",
    "serial": "C35E0C1F5CE9414300011018",
    "size": 1073741824,
    "source": "v1"
  },
  {
    "created": "2017-10-06T22:00:27Z",
    "name": "pg2.1.v1",
    "serial": "C35E0C1F5CE9414300011019",
    "size": 1073741824,
    "source": "v1"
  }
]
```

3.3 GET volume/{volume}/diff

Lists block differences for the specified volume snapshot.

Parameters

Parameter	Type	Description
base	string	Snapshot name to be used as the base for the diff. If a base snapshot is not specified, all mapped blocks for the volume are returned.

		Optional.
block_size	number or string	Granularity, in bytes, at which to compare. Required.
length	number or string	Length of the region, in bytes, to compare. Required.
offset	number or string	Absolute offset, in bytes, of the region to compare. Must be a multiple of block_size. Optional.

Example 1: List all block differences for volume snapshot v3.snap1 with a block size of 512 bytes and a length of 2 gigabytes.

Request:

```
GET https://pure01.example.com/api/1.13/volume/v3.snap1/diff?block_size=512&length=2G
```

Response:

```
[]
```

3.4 GET volume/{volume}/hgroup

Lists shared connections for the specified volume.

Parameters

Parameter	Type	Description
None	n/a	n/a

Example 1: List shared connections for volume v2

Request:

```
GET https://pure01.example.com/api/1.13/volume/v2/hgroup
```

Response:

```
[
  {
    "hgroup": "hg2",
    "host": null,
    "lun": 15,
    "name": "v2",
    "size": 2147483648
  }
]
```

3.5 GET volume/{volume}/host

Lists private connections for the specified volume.

Parameters

Parameter	Type	Description
None	n/a	n/a

Example 1: List private connections for volume v3

Request:

```
GET https://pure01.example.com/api/1.13/volume/v3/host
```

Response:

```
[
  {
    "host": "h2",
    "lun": 7,
    "name": "v3",
    "size": 3221225472
  }
]
```

3.6 POST volume/{volume}

Creates a volume or copies a volume or snapshot.

Either the `size` or `source` parameter must be specified.

Parameters

Parameter	Type	Description
overwrite	boolean	When used with `source`, overwrites (`true`) an existing volume. Optional.
size	number or string	Creates a volume with the specified provisioned size. Enter the size as a number (bytes) or as a string with a single character unit symbol. Valid unit symbols are `K`, `M`, `G`, `T`, `P`, representing KiB, MiB, GiB, TiB, and PiB, respectively, where "Ki" denotes 2^10, "Mi" denotes 2^20, and so on. If the unit symbol is not specified, the unit defaults to bytes. Optional.
source	string	Copies a volume or snapshot to create a new volume or replace an existing one. Specify the name of a volume or snapshot whose data is copied to the volume specified. If the volume or snapshot replaces an existing volume, the `overwrite` parameter must also be specified. If the volume or snapshot replaces an existing volume, an undo snapshot is automatically taken (providing a 24-hour window during which the previous contents can be retrieved). Optional.

Example 1: Create a new volume

Request:

```
POST https://pure01.example.com/api/1.13/volume/v5
{
  "size": "5G"
}
```

Response:

```
{
  "created": "2017-10-06T22:00:34Z",
  "name": "v5",
  "serial": "C35E0C1F5CE941430001101E",
  "size": 5368709120,
  "source": null
}
```

Example 2: Create v6 by copying v1

Request:

```
POST https://pure01.example.com/api/1.13/volume/v6
{
  "source": "v1"
}
```

Response:

```
{
  "created": "2017-10-06T22:00:34Z",
  "name": "v6",
  "serial": "C35E0C1F5CE941430001101F",
  "size": 1073741824,
  "source": "v1"
}
```

Example 3: Overwrite v6 by copying v1

Request:

```
POST https://pure01.example.com/api/1.13/volume/v6
{
  "overwrite": true,
  "source": "v1"
}
```

Response:

```
{
  "created": "2017-10-06T22:00:34Z",
  "name": "v6",
  "serial": "C35E0C1F5CE941430001101F",
  "size": 1073741824,
  "source": "v1"
}
```

3.7 POST volume

Creates snapshots of one or more volumes.

Parameters

Parameter	Type	Description
snap	boolean	Must be set to `true` to signify that a snapshot is being created. Required.
source	list	Names of one or more volumes to snapshot. Required.
suffix	string	Specify a custom suffix that is added to the snapshot name. Optional.

Example 1: Create snapshots of v5 and v6

Request:

```
POST https://pure01.example.com/api/1.13/volume
{
  "snap": true,
  "source": [
    "v5",
    "v6",
```

```
  ]
}
```

Response:

```
[
  {
    "created": "2017-10-06T22:00:34Z",
    "name": "v5.1",
    "serial": "C35E0C1F5CE9414300011021",
    "size": 5368709120,
    "source": "v5"
  },
  {
    "created": "2017-10-06T22:00:34Z",
    "name": "v6.4130",
    "serial": "C35E0C1F5CE9414300011022",
    "size": 1073741824,
    "source": "v6"
  }
]
```

Example 2: Create snapshots of v5 and v6 with a custom suffix

Request:

```
POST https://pure01.example.com/api/1.13/volume
{
  "snap": true,
  "source": [
    "v5",
    "v6"
  ],
  "suffix": "crackle"
}
```

Response:

```
[
  {
    "created": "2017-10-06T22:00:35Z",
    "name": "v5.crackle",
    "serial": "C35E0C1F5CE9414300011023",
    "size": 5368709120,
    "source": "v5"
  },
  {
    "created": "2017-10-06T22:00:35Z",
    "name": "v6.crackle",
    "serial": "C35E0C1F5CE9414300011024",
    "size": 1073741824,
    "source": "v6"
  }
]
```

3.8 POST volume/{volume}/pgroup/{pgroup}

Adds a volume to a protection group.

You can only add members of the same object type to a protection group. For example, you cannot add hosts or host groups to a protection group that contains volumes.

Parameters

Parameter	Type	Description
None	n/a	n/a

3.9 PUT volume/{volume}

Recovers and renames the specified volume or volume snapshot, or resizes the specified volume.

Either the `size`, `action` or `name` parameter must be specified.

Parameters

Parameter	Type	Description
action	string	Recovers the contents of the specified volume or volume snapshot. Set the parameter to recover. Optional.
name	string	Renames the specified volume or volume snapshot. When renaming a snapshot, only the suffix can be changed. Optional.
size	number or string	Changes the provisioned size of the volume. If the new volume size is smaller than the original, then the `truncate` parameter needs to be set to `true`. Enter the size as a number (bytes) or as a string with a single character unit symbol. Valid unit symbols are `K`, `M`, `G`, `T`, `P`, representing KiB, MiB, GiB, TiB, and PiB, respectively, where "Ki" denotes 2^10, "Mi" denotes 2^20, and so on. If the unit symbol is not specified, the unit defaults to bytes. Optional.
truncate	boolean	Truncates (`true`) the volume. When a volume is truncated, Purity automatically takes an undo snapshot, providing a 24-hour window during which the previous contents can be retrieved. After truncating a volume, its provisioned size can be subsequently increased, but the data in truncated sectors cannot be retrieved. Required if the volume `size` is set to a size that is smaller than the original.

Example 1: Set the provisioned size for volume v5 to 10G

Request:

```
PUT https://pure01.example.com/api/1.13/volume/v5
{
  "size": "10G"
}
```

Response:

```
{
  "name": "v5",
  "size": 10737418240
}
```

Example 2: Truncate volume v5 to 1073741824 bytes

Request:

```
PUT https://pure01.example.com/api/1.13/volume/v5
{
  "size": 1073741824,
  "truncate": true
}
```

Response:

```
{
  "name": "v5",
  "size": 1073741824
}
```

Example 3: Rename volume v5 to v5_renamed

Request:

```
PUT https://pure01.example.com/api/1.13/volume/v5
{
  "name": "v5_renamed"
}
```

Response:

```
{
  "name": "v5_renamed"
}
```

Example 4: Rename suffix of volume snapshot v5.snap from snap to snap2

Request:

```
PUT https://pure01.example.com/api/1.13/volume/v3.snap1
{
  "name": "v3.snap2"
}
```

Response:

```
{
  "name": "v3.snap2"
}
```

3.10 DELETE volume/{volume}

Destroys or eradicates the specified volume or snapshot.

Parameters

Parameter	Type	Description
eradicate	boolean	If set to `true`, eradicates the specified volume or snapshot. If destroying or eradicating a volume, its snapshots are also destroyed or eradicated. After destroying a volume or snapshot, you can eradicate it to immediately terminate the 24-hour eradication period and begin storage reclamation. Once eradication has begun, the volume or snapshot can no longer be recovered. If set to `false`, destroys the specified volume or snapshot. The destroyed volume or snapshot undergoes a 24-hour eradication pending period during which time the volume or snapshot and its data can be fully recovered. After the 24-hour pending period, Purity eradicates the destroyed volume or snapshot. Optional.

Example 1: Destroy volume v6

Request:

```
DELETE https://pure01.example.com/api/1.13/volume/v6
```

Response:

```
{
  "name": "v6"
}
```

Example 2: Eradicate volume v6

Request:

```
DELETE https://pure01.example.com/api/1.13/volume/v6?eradicate=true
```

Response:

```
{
  "name": "v6"
}
```

3.11 DELETE volume/{volume}/pgroup/{pgroup}

Removes a volume member from a protection group.

Parameters

Parameter	Type	Description
None	n/a	n/a

4. Hosts

View, create, and manage hosts.

4.1 GET host

Lists all hosts on the array.

Parameters

Parameter	Type	Description
action	string	If set to `monitor`, displays host-wide I/O performance information in real time. Time is displayed in Greenwich Mean Time (GMT). Optional.
all	boolean	Displays all visible attributes of the specified hosts, including associated worldwide names, host groups, connected volumes and the LUNs used to address them, and array port worldwide names through which the volumes are visible. Optional.
chap	boolean	Displays host and target user names and indicates whether host and target passwords have been set. Optional.
connect	boolean	Displays all host-volume connections, either private or shared, and the LUNs used by the hosts to address the volumes. Optional.
names	list	Displays specified hosts. Enter multiple host names in comma-separated format. Host names can include asterisk (`*`) wildcard characters to represent zero or more characters. Optional.
personality	boolean	Displays the personality setting associated with the specified hosts. Optional.
private	boolean	Used with the `connect` parameter to display (`true`) private host-volume connections. Optional.
protect	boolean	Displays all protected hosts and their associated protection groups. Optional.

shared	boolean	Used with the `connect` parameter to display (`true`) shared host-volume connections. Optional.
space	boolean	Displays information about provisioned (virtual) size and physical storage consumption for each volume connected to the specified hosts. Optional.

Example 1: List all hosts

Request:

```
GET https://pure01.example.com/api/1.13/host
```

Response:

```
[
  {
    "hgroup": "hg1",
    "iqn": [],
    "name": "h1",
    "wwn": [
      "0000111122223333"
    ]
  },
  {
    "hgroup": "hg1",
    "iqn": [],
    "name": "h2",
    "wwn": []
  },
  {
    "hgroup": null,
    "iqn": [],
    "name": "h3",
    "wwn": []
  }
]
```

Example 2: Display I/O performance for all hosts

Request:

```
GET https://pure01.example.com/api/1.13/host?action=monitor
```

Response:

```
[
  {
    "input_per_sec": 0,
    "name": "h1",
    "output_per_sec": 0,
    "reads_per_sec": 0,
    "san_usec_per_read_op": 0,
    "san_usec_per_write_op": 0,
    "time": "2017-10-06T22:00:37Z",
    "usec_per_read_op": 0,
    "usec_per_write_op": 0,
    "writes_per_sec": 0
  },
  {
    "input_per_sec": 0,
    "name": "h2",
    "output_per_sec": 0,
    "reads_per_sec": 0,
    "san_usec_per_read_op": 0,
    "san_usec_per_write_op": 0,
    "time": "2017-10-06T22:00:37Z",
    "usec_per_read_op": 0,
    "usec_per_write_op": 0,
    "writes_per_sec": 0
  },
...
    "usec_per_write_op": 0,
    "writes_per_sec": 0
  }
```

```
]
```

4.2 GET host/{host}

Lists the attributes for the specified host.

Parameters

Parameter	Type	Description
action	string	If set to `monitor`, displays host-wide I/O performance information in real time. Time is displayed in Greenwich Mean Time (GMT). Optional.
all	boolean	Displays all visible attributes of the specified host. Display includes associated worldwide names, host groups, connected volumes and the LUNs used to address them, array port worldwide names through which the volumes are visible. Optional.
chap	boolean	Displays host and target usernames and indicates whether host and target passwords have been set. Optional.
personality	boolean	Displays the personality setting associated with the specified host. Optional.
protect	boolean	Displays the names of all protection groups associated with the specified host. If the host is not protected, returns an empty response. Optional.
space	boolean	Displays information about provisioned (virtual) size and physical storage consumption for each volume connected to the specified host. Optional.

Example 1: List space usage for host h1

Request:

```
GET https://pure01.example.com/api/1.13/host/h1?space=true
```

Response:

```
{
  "data_reduction": 1.0,
  "name": "h1",
  "size": 1073741824,
  "snapshots": 0,
  "thin_provisioning": 1.0,
  "total": 0,
  "total_reduction": 1.0,
  "volumes": 0
}
```

Example 2: List CHAP attributes for host h1

Request:

```
GET https://pure01.example.com/api/1.13/host/h1?chap=true
```

Response:

```
{
  "host_password": null,
  "host_user": null,
  "name": "h1",
  "target_password": null,
  "target_user": null
}
```

Example 3: List personality attribute for host h1

Request:

```
GET https://pure01.example.com/api/1.13/host/h1?personality=true
```

Response:

```
{
  "name": "h1",
  "personality": null
}
```

Example 4: Display I/O performance for host h1

Request:

```
GET https://pure01.example.com/api/1.13/host/h1?action=monitor
```

Response:

```
{
  "input_per_sec": 0,
  "name": "h1",
  "output_per_sec": 0,
  "reads_per_sec": 0,
  "san_usec_per_read_op": 0,
  "san_usec_per_write_op": 0,
  "time": "2017-10-06T22:00:37Z",
  "usec_per_read_op": 0,
  "usec_per_write_op": 0,
  "writes_per_sec": 0
}
```

4.3 GET host/{host}/volume

Lists volumes associated with the specified host and the LUNs used to address them.

Parameters

Parameter	Type	Description
private	boolean	Restricts (`true`) the display list to volumes with private connections. Optional.
shared	boolean	Restricts (`true`) the display list to volumes with shared connections. Optional.

Example 1: List private and shared connections for host h2

Request:

```
GET https://pure01.example.com/api/1.13/host/h2/volume
```

Response:

```
[
  {
    "hgroup": "hg1",
    "lun": 254,
    "name": "h2",
    "vol": "v1"
  },
  {
    "hgroup": null,
    "lun": 7,
    "name": "h2",
    "vol": "v3"
  }
]
```

Example 2: List private connections for host h2

Request:

```
GET https://pure01.example.com/api/1.13/host/h2/volume?private=true
```

Response:

```
[
  {
    "lun": 7,
    "name": "h2",
    "vol": "v3"
  }
]
```

Example 3: List shared connections for host h2

Request:

```
GET https://pure01.example.com/api/1.13/host/h2/volume?shared=true
```

Response:

```
[
  {
    "hgroup": "hg1",
    "lun": 254,
    "name": "h2",
    "vol": "v1"
  }
]
```

4.4 POST host/{host}

Creates a host with the specified name.

Parameters

Parameter	Type	Description
iqnlist	list	Sets the list of iSCSI qualified names (IQNs) for the new host. Optional.
wwnlist	list	Sets the list of Fibre Channel worldwide names (WWNs) for the new host. Optional.

Example 1: Create host h4 with one wwn

Request:

```
POST https://pure01.example.com/api/1.13/host/h4
{
  "wwnlist": [
    "0000999900009999"
  ]
}
```

Response:

```
{
  "iqn": [],
  "name": "h4",
  "wwn": [
    "0000999900009999"
  ]
}
```

4.5 POST host/{host}/pgroup/{pgroup}

Adds a host to a protection group.

You can only add members of the same object type to a protection group. For example, you cannot add hosts or host groups to a protection group that contains volumes.

Parameters

Parameter	Type	Description
None	n/a	n/a

4.6 POST host/{host}/volume/{volume}

Connects a volume to the host.

Parameters

Parameter	Type	Description
lun	number	Assigns the same LUN to each connection. The connection fails for any host for which the specified LUN is already in use. Optional.

Example 1: Create a private connection between host h4 and volume v5_renamed

Request:

```
POST https://pure01.example.com/api/1.13/host/h4/volume/v5_renamed
```

Response:

```
{
  "lun": 1,
  "name": "h4",
  "vol": "v5_renamed"
}
```

4.7 PUT host/{host}

Modifies the attributes for the host.

At least one parameter must be specified.

Parameters

Parameter	Type	Description
addiqnlist	list	Adds the list of iSCSI qualified names (IQNs) to the specified host. Optional.
addwwnlist	list	Adds the list of Fibre Channel worldwide names (WWNs) to the specified host. Optional.
host_password	string	Sets the host password for CHAP authentication. Optional.
host_user	string	Sets the host user name for CHAP authentication. Optional.
iqnlist	list	Assigns a list of iSCSI qualified names (IQNs) to the specified host. Optional.
name	string	Changes the name of the host to the specified name. Optional.
personality	string	Determines the way in which the host personality tunes the protocol used between the array and the initiator. If the host is running the HP-UX operating system, set the host personality to 'hpux'. Optional.
remiqnlist	list	Removes the list of iSCSI qualified names (IQNs) from the specified host. Optional.
remwwnlist	list	Removes the list of Fibre Channel worldwide names (WWNs) from the specified host. Optional.
target_password	string	Sets the target password for CHAP authentication. Optional.
target_user	string	Sets the target user name for CHAP authentication. Optional.
wwnlist	list	Assigns a list of Fibre Channel worldwide names (WWNs) to the specified host. Optional.

Example 1: Replace list of wwns

Request:

```
PUT https://pure01.example.com/api/1.13/host/h4
{
  "wwnlist": [
    "1111222233334444",
    "2222333344445555"
  ]
}
```

Response:

```
{
  "hgroup": null,
  "iqn": [],
  "name": "h4",
  "wwn": [
    "1111222233334444",
    "2222333344445555"
```

```
  ]
}
```

Example 2: Add two more wwns to list

Request:

```
PUT https://pure01.example.com/api/1.13/host/h4
{
  "addwwnlist": [
    "4444333322221111",
    "5555444433332222"
  ]
}
```

Response:

```
{
  "hgroup": null,
  "iqn": [],
  "name": "h4",
  "wwn": [
    "1111222233334444",
    "2222333344445555",
    "4444333322221111",
    "5555444433332222"
  ]
}
```

Example 3: Remove one wwn from list

Request:

```
PUT https://pure01.example.com/api/1.13/host/h4
{
  "remwwnlist": [
    "2222333344445555"
  ]
}
```

Response:

```
{
  "hgroup": null,
  "iqn": [],
  "name": "h4",
  "wwn": [
    "1111222233334444",
    "4444333322221111",
    "5555444433332222"
  ]
}
```

Example 4: Rename a host

Request:

```
PUT https://pure01.example.com/api/1.13/host/h4
{
  "name": "h5"
}
```

Response:

```
{
  "name": "h5"
}
```

4.8 DELETE host/{host}/pgroup/{pgroup}

Removes a host member from a protection group.

Parameters

Parameter	Type	Description
None	n/a	n/a

4.9 DELETE host/{host}/volume/{volume}

Breaks the connection between a host and volume.

Parameters

Parameter	Type	Description
None	n/a	n/a

Example 1: Disconnect host h5 and volume v5_renamed

Request:

```
DELETE https://pure01.example.com/api/1.13/host/h5/volume/v5_renamed
```

Response:

```
{
  "name": "h5",
  "vol": "v5_renamed"
}
```

4.10 DELETE host/{host}

Removes the specified host.

Hosts cannot be deleted while they have private connections to volumes or while they are associated with host groups.

Parameters

Parameter	Type	Description
None	n/a	n/a

Example 1: Delete host h5

Request:

```
DELETE https://pure01.example.com/api/1.13/host/h5
```

Response:

```
{
  "name": "h5"
}
```

5. Host Groups

View, create, and manage host groups.

5.1 GET hgroup

Lists or monitors all host groups.

Parameters

Parameter	Type	Description
action	string	If set to `monitor`, displays host group I/O performance information for all host groups. Optional.
connect	boolean	Displays all volumes associated with host groups, and the LUNs used by the associated hosts to address them. Optional.
names	list	Displays specified host groups. Enter multiple host group names in comma-separated format. Host group names can include asterisk (`*`) wildcard characters to represent zero or more characters. Optional.
protect	boolean	Displays (`true`) all protected host groups and their associated protection groups. Optional.
space	boolean	Displays size and space consumption information for each volume associated with each specified host group. Optional.

Example 1: List all host groups

Request:

```
GET https://pure01.example.com/api/1.13/hgroup
```

Response:

```
[
  {
    "hosts": [
      "h1",
      "h2"
    ],
    "name": "hg1"
  },
  {
    "hosts": [],
    "name": "hg2"
  }
]
```

Example 2: List all host groups' space usage

Request:

```
GET https://pure01.example.com/api/1.13/hgroup?space=true
```

Response:

```
[
  {
    "data_reduction": 1.0,
    "name": "hg1",
    "size": 1073741824,
    "snapshots": 0,
    "thin_provisioning": 1.0,
    "total": 0,
    "total_reduction": 1.0,
    "volumes": 0
  },
  {
    "data_reduction": 1.0,
```

```
    "name": "hg2",
    "size": 2147483648,
    "snapshots": 0,
    "thin_provisioning": 1.0,
    "total": 0,
    "total_reduction": 1.0,
    "volumes": 0
  }
]
```

Example 3: Monitor I/O performance of all host groups

Request:

```
GET https://pure01.example.com/api/1.13/hgroup?action=monitor
```

Response:

```
[
  {
    "input_per_sec": 0,
    "name": "hg2",
    "output_per_sec": 0,
    "reads_per_sec": 0,
    "san_usec_per_read_op": 0,
    "san_usec_per_write_op": 0,
    "time": "2017-10-06T22:00:41Z",
    "usec_per_read_op": 0,
    "usec_per_write_op": 0,
    "writes_per_sec": 0
  },
  {
    "input_per_sec": 0,
    "name": "hg1",
    "output_per_sec": 0,
    "reads_per_sec": 0,
    "san_usec_per_read_op": 0,
    "san_usec_per_write_op": 0,
    "time": "2017-10-06T22:00:41Z",
    "usec_per_read_op": 0,
    "usec_per_write_op": 0,
    "writes_per_sec": 0
  }
]
```

5.2 GET hgroup/{hgroup}

Lists or monitors a single host group.

Parameters

Parameter	Type	Description
action	string	If set to `monitor`, displays host group I/O performance information for the specified host group. Optional.
protect	boolean	Displays (`true`) the names of all protection groups associated with the specified host group. If the host group is not protected, returns an empty response. Optional.
space	boolean	Displays size and space consumption information for each volume associated with the specified host group. Optional.

Example 1: List host group hg1

Request:

```
GET https://pure01.example.com/api/1.13/hgroup/hg1
```

Response:

```
{
  "hosts": [
    "h1",
    "h2"
  ],
  "name": "hg1"
}
```

Example 2: List space usage for host group hg1

Request:

```
GET https://pure01.example.com/api/1.13/hgroup/hg1?space=true
```

Response:

```
{
  "data_reduction": 1.0,
  "name": "hg1",
  "size": 1073741824,
  "snapshots": 0,
  "thin_provisioning": 1.0,
  "total": 0,
  "total_reduction": 1.0,
  "volumes": 0
}
```

Example 3: Monitor I/O performance for host group hg1

Request:

```
GET https://pure01.example.com/api/1.13/hgroup/hg1?action=monitor
```

Response:

```
{
  "input_per_sec": 0,
  "name": "hg1",
  "output_per_sec": 0,
  "reads_per_sec": 0,
  "san_usec_per_read_op": 0,
  "san_usec_per_write_op": 0,
  "time": "2017-10-06T22:00:41Z",
  "usec_per_read_op": 0,
  "usec_per_write_op": 0,
  "writes_per_sec": 0
}
```

5.3 GET hgroup/{hgroup}/volume

Lists volumes associated with the specified host groups and the LUNs used to address them.

Parameters

Parameter	Type	Description
None	n/a	n/a

Example 1: List volumes connected to host group hg1

Request:

```
GET https://pure01.example.com/api/1.13/hgroup/hg1/volume
```

Response:

```
[
  {
    "lun": 254,
    "name": "hg1",
    "vol": "v1"
  }
]
```

5.4 POST hgroup/{hgroup}

Creates a host group with the specified name.

Parameters

Parameter	Type	Description
hostlist	list	List of member hosts. Optional.

Example 1: Create host group hg3

Request:

```
POST https://pure01.example.com/api/1.13/hgroup/hg3
```

Response:

```
{
  "hosts": [],
  "name": "hg3"
}
```

5.5 POST hgroup/{hgroup}/pgroup/{pgroup}

Adds a host group to a protection group.

You can only add members of the same object type to a protection group. For example, you cannot add hosts or host groups to a protection group that contains volumes.

Parameters

Parameter	Type	Description
None	n/a	n/a

5.6 POST hgroup/{hgroup}/volume/{volume}

Connects a volume to all hosts in the specified host group.

If the LUN is not specified, when the volume is connected to the host group, Purity assigns the same LUN to each connection. All hosts in the group use this LUN to communicate with the volume.

Parameters

Parameter	Type	Description
lun	number	Assigns the specified LUN to each connection. Optional.

Example 1: Connect volume v3 to host group hg3

Request:

```
POST https://pure01.example.com/api/1.13/hgroup/hg3/volume/v3
```

Response:

```
{
  "lun": 254,
  "name": "hg3",
  "vol": "v3"
}
```

5.7 PUT hgroup/{hgroup}

Adds, removes, or modifies host members of the host group.

At least one parameter must be specified.

Parameters

Parameter	Type	Description
addhostlist	list	Adds a list of hosts to the existing list. Optional.
hostlist	list	Replaces the list of member hosts. Optional.
name	string	Changes the name of the host group to the specified name. Optional.
remhostlist	list	Removes the list of hosts from the existing list. Optional.

Example 1: Rename host group

Request:

```
PUT https://pure01.example.com/api/1.13/hgroup/hg3
{
  "name": "hg4"
}
```

Response:

```
{
  "hosts": [],
  "name": "hg4"
}
```

Example 2: Set host members for hgroup hg4

Request:

```
PUT https://pure01.example.com/api/1.13/hgroup/hg4
{
  "hostlist": [
    "h3"
  ]
}
```

Response:

```
{
  "hosts": [
    "h3"
  ],
  "name": "hg4"
}
```

Example 3: Remove host members

Request:

```
PUT https://pure01.example.com/api/1.13/hgroup/hg4
{
  "remhostlist": [
    "h3"
  ]
}
```

Response:

```
{
  "hosts": [],
  "name": "hg4"
}
```

5.8 DELETE hgroup/{hgroup}/pgroup/{pgroup}

Removes a host group member from a protection group.

Parameters

Parameter	Type	Description
None	n/a	n/a

5.9 DELETE hgroup/{hgroup}/volume/{volume}

Breaks the connections between all hosts associated with the host group and the specified volume.

Parameters

Parameter	Type	Description
None	n/a	n/a

Example 1: Disconnect volume v3 from host group hg4

Request:

```
DELETE https://pure01.example.com/api/1.13/hgroup/hg4/volume/v3
```

Response:

```
{
  "name": "hg4",
  "vol": "v3"
}
```

5.10 DELETE hgroup/{hgroup}

Deletes the specified host group. You must disconnect all volumes and remove all hosts from the host group before deleting the host group.

Parameters

Parameter	Type	Description
None	n/a	n/a

Example 1: Delete host group hg4

Request:

```
DELETE https://pure01.example.com/api/1.13/hgroup/hg4
```

Response:

```
{
  "name": "hg4"
}
```

6. Protection Groups

View, create, and manage protection groups.

6.1 GET pgroup

Lists protection groups and protection group snapshots and their attributes.

Parameters

Parameter	Type	Description
names	list	Displays specified protection groups. Enter multiple protection group names in comma-separated format. Protection group names can include asterisk (`*`) wildcard characters to represent zero or more characters. Optional.
pending	boolean	Includes (`true`) destroyed protection groups that are in the eradication pending state. Time remaining is displayed in seconds. The `pending` option can also be used with the `schedule`, `retention`, `space`, or `snap` options to include a list of pending protection groups or snapshots. Optional.
pending_only	boolean	Lists (`true`) destroyed protection groups that are in the eradication pending state. Time remaining is displayed in seconds. The `pending` option can also be used with the `schedule`, `retention`, `space`, or `snap` options to include a list of pending protection groups or snapshots. Optional.
retention	boolean	Displays (`true`) the source/target retention policy. Optional.
schedule	boolean	Displays (`true`) the snapshot/replication schedule. Optional.
snap	boolean	Lists all snapshots (`true`). Can be used with the `space` parameter to list space consumption for all snapshots. Optional.
source	boolean	Lists (`true`) protection groups and snapshots created on this array. The `source` option can be used with the `schedule`, `retention`, `space`, or `snap` options. Optional.

space	boolean	Displays (`true`) size and space consumption information for each volume associated with each specified protection group. Optional.
target	boolean	Lists (`true`) protection groups and snapshots replicated to this array. The `target` option can be used with the `schedule`, `retention`, `space`, or `snap` options. Optional.
total	boolean	Displays (`true`) the total space consumption for all protection groups. Must be used with the `space` parameter. Optional.
transfer	boolean	Used with the `snap` parameter to display (`true`) replication data transfer statistics, including data transfer start time, data transfer end time, data transfer progress, and amount of logical/physical data transferred. Optional.

Example 1: List all protection groups.

Request:

```
GET https://pure01.example.com/api/1.13/pgroup
```

Response:

```
[
  {
    "hgroups": null,
    "hosts": null,
    "name": "pg1",
    "source": "pure01",
    "targets": null,
    "volumes": [
      "v1"
    ]
  },
  {
    "hgroups": null,
    "hosts": null,
    "name": "pg2",
    "source": "pure01",
    "targets": null,
    "volumes": [
      "v1"
    ]
  }
]
```

Example 2: List all protection groups created on this array.

Request:

```
GET https://pure01.example.com/api/1.13/pgroup?source=true
```

Response:

```
[
  {
    "hgroups": null,
    "hosts": null,
    "name": "pg1",
    "source": "pure01",
    "targets": null,
    "volumes": [
      "v1"
    ]
  },
  {
    "hgroups": null,
    "hosts": null,
    "name": "pg2",
    "source": "pure01",
```

```
    "targets": null,
    "volumes": [
      "v1"
    ]
  }
]
```

Example 3: List all protection groups and their schedules.

Request:

```
GET https://pure01.example.com/api/1.13/pgroup?schedule=true
```

Response:

```
[
  {
    "name": "pg1",
    "replicate_at": null,
    "replicate_blackout": null,
    "replicate_enabled": false,
    "replicate_frequency": 14400,
    "snap_at": null,
    "snap_enabled": false,
    "snap_frequency": 3600
  },
  {
    "name": "pg2",
    "replicate_at": null,
    "replicate_blackout": null,
    "replicate_enabled": false,
    "replicate_frequency": 14400,
    "snap_at": null,
    "snap_enabled": false,
    "snap_frequency": 3600
  }
]
```

Example 4: List all protection groups and their retention policies.

Request:

```
GET https://pure01.example.com/api/1.13/pgroup?retention=true
```

Response:

```
[
  {
    "all_for": 86400,
    "days": 7,
    "name": "pg1",
    "per_day": 4,
    "target_all_for": 86400,
    "target_days": 7,
    "target_per_day": 4
  },
  {
    "all_for": 86400,
    "days": 7,
    "name": "pg2",
    "per_day": 4,
    "target_all_for": 86400,
    "target_days": 7,
    "target_per_day": 4
  }
]
```

Example 5: List all protection groups including protection pgroups in the eradication pending state.

Request:

```
GET https://pure01.example.com/api/1.13/pgroup?pending=true
```

Response:

```
[
  {
    "hgroups": null,
    "hosts": null,
    "name": "pg1",
    "source": "pure01",
    "targets": null,
    "time_remaining": null,
    "volumes": [
      "v1"
    ]
  },
  {
    "hgroups": null,
    "hosts": null,
    "name": "pg2",
    "source": "pure01",
    "targets": null,
    "time_remaining": null,
    "volumes": [
      "v1"
    ]
  },
  {
    "hgroups": null,
...
      "v3"
    ]
  }
]
```

Example 6: List all protection groups in the eradication pending state.

Request:

```
GET https://pure01.example.com/api/1.13/pgroup?pending_only=true
```

Response:

```
[
  {
    "hgroups": null,
    "hosts": null,
    "name": "pg3",
    "source": "pure01",
    "targets": null,
    "time_remaining": 86384,
    "volumes": [
      "v2",
      "v3"
    ]
  }
]
```

Example 7: List all protection groups and their space usage.

Request:

```
GET https://pure01.example.com/api/1.13/pgroup?space=true
```

Response:

```
[
  {
    "name": "pg1",
    "snapshots": 0
  },
  {
    "name": "pg2",
    "snapshots": 0
  }
]
```

6.2 GET pgroup/{pgroup}

Lists the attributes for the specified protection group or protection group snapshot.

If listing protection group snapshot attributes, the `snap` parameter must be specified.

Parameters

Parameter	Type	Description
pending	boolean	Includes (`true`) destroyed protection groups that are in the eradication pending state. Time remaining is displayed in seconds. The `pending` option can also be used with the `schedule`, `retention`, `space`, or `snap` options to include a list of pending protection groups or snapshots. Optional.
pending_only	boolean	Used with the `snap` parameter to list (`true`) destroyed protection group snapshots that are in the eradication pending state. Time remaining is displayed in seconds. Optional.
retention	boolean	Displays (`true`) the source/target retention policy. Optional.
schedule	boolean	Displays (`true`) the snapshot/replication schedule. Optional.
snap	boolean	If listing the attributes for a specified protection group, set `snap` to `true` to list all snapshots within the protection group. If listing the attributes for a specified protection group snapshot, always set `snap` to `true`. Can be used with the `space` parameter to list space consumption for all snapshots. Optional if listing attributes for a specified protection group. Required if listing attributes for a specified protection group snapshot.
space	boolean	Displays (`true`) size and space consumption information for each volume associated with each specified protection group. Optional.
total	boolean	Used with the `snap` parameter to display (`true`) the total space consumption for all snapshots of this protection group. Must be used with the `space` parameter. Optional.
transfer	boolean	Used with the `snap` parameter to display (`true`) replication data transfer statistics, including data transfer start time, data transfer end time, data transfer progress, and amount of logical/physical data transferred. Optional.

Example 1: List protection group pg1.

Request:

```
GET https://pure01.example.com/api/1.13/pgroup/pg1
```

Response:

```
{
  "hgroups": null,
  "hosts": null,
  "name": "pg1",
  "source": "pure01",
  "targets": null,
  "volumes": [
    "v1"
  ]
}
```

Example 2: List protection group pg3 even if it is in eradication pending state.

Request:

```
GET https://pure01.example.com/api/1.13/pgroup/pg3?pending=true
```

Response:

```
{
  "hgroups": null,
  "hosts": null,
  "name": "pg3",
  "source": "pure01",
  "targets": null,
  "time_remaining": 86384,
  "volumes": [
    "v2",
    "v3"
  ]
}
```

Example 3: List space usage for protection group pg1.

Request:

```
GET https://pure01.example.com/api/1.13/pgroup/pg1?space=true
```

Response:

```
{
  "name": "pg1",
  "snapshots": 0
}
```

Example 4: List schedule for protection group pg1.

Request:

```
GET https://pure01.example.com/api/1.13/pgroup/pg1?schedule=true
```

Response:

```
{
  "name": "pg1",
  "replicate_at": null,
  "replicate_blackout": null,
  "replicate_enabled": false,
  "replicate_frequency": 14400,
  "snap_at": null,
  "snap_enabled": false,
  "snap_frequency": 3600
}
```

6.3 POST pgroup

Creates snapshots of one or more protection groups.

Parameters

Parameter	Type	Description
apply_retention	boolean	Applies the retention schedule settings of the protection group to the snapshot. Optional.
replicate	boolean	Queues this snapshot for replication to all allowed target arrays. Purity will begin replicating data to each allowed target when all earlier replication sessions for the same protection group have been completed to that target, excluding those started with `replicate_now`. May not be used with `replicate_now`.

		Optional.
replicate_now	boolean	Replicates this snapshot immediately to all allowed target arrays. May not be used with `replicate`. Optional.
snap	boolean	Must be set to `true` to signify that a snapshot is being created. Required.
source	list	Generates a snapshot for each of the protection groups specified. Required.
suffix	string	Specifies a name suffix for the snapshot to be created. Optional.

Example 1: Create a snapshot of protection group pg1.

Request:

```
POST https://pure01.example.com/api/1.13/pgroup
{
  "snap": true,
  "source": [
    "pg1"
  ]
}
```

Response:

```
[
  {
    "created": "2017-10-06T22:00:44Z",
    "name": "pg1.3",
    "source": "pg1"
  }
]
```

Example 2: Create a snapshot of protection group pg1with a custom suffix and apply pg1's retention policy to it.

Request:

```
POST https://pure01.example.com/api/1.13/pgroup
{
  "apply_retention": true,
  "snap": true,
  "source": [
    "pg1"
  ],
  "suffix": "crackle"
}
```

Response:

```
[
  {
    "created": "2017-10-06T22:00:44Z",
    "name": "pg1.crackle",
    "source": "pg1"
  }
]
```

6.4 POST pgroup/{pgroup}

Creates a protection group with the specified name or copies a protection group.

When copying a protection group, the destination protection group is created if it did not exist previously.

Parameters

Parameter	Type	Description
hgrouplist	list	List of one or more host groups to be included in the new protection group. Optional.
hostlist	list	List of one or more hosts to be included in the new protection group. Optional.
overwrite	boolean	When copying a protection group, overwrite (`true`) volumes in the destination protection group that have the same name as the volumes in the source protection group. Without the `overwrite=true` parameter, the copy command fails if any volume name collisions are found. With the `overwrite=true` parameter, both the source and destination protection groups must contain exactly the same volumes (volume names), otherwise the entire copy command fails. Optional.
source	string	The source pgroup or pgroup snapshot to copy from. Optional.
targetlist	list	List of one or more targets to be included in the new protection group. Optional.
vollist	list	List of one or more volumes to be included in the new protection group. Optional.

Example 1: Create protection group pg4.

Request:

```
POST https://pure01.example.com/api/1.13/pgroup/pg4
```

Response:

```
{
  "hgroups": null,
  "hosts": null,
  "name": "pg4",
  "source": "pure01",
  "targets": null,
  "volumes": null
}
```

Example 2: Create protection group pg5 with host h1.

Request:

```
POST https://pure01.example.com/api/1.13/pgroup/pg5
{
  "hostlist": [
    "h1"
  ]
}
```

Response:

```
{
  "hgroups": null,
  "hosts": [
    "h1"
  ],
  "name": "pg5",
  "source": "pure01",
  "targets": null,
  "volumes": null
}
```

6.5 PUT pgroup/{pgroup}

Modifies a protection group, its snapshot/replication schedule or its retention policy.

At least one parameter must be specified.

Parameters

Parameter	Type	Description
action	string	If set to `"recover"`, recovers a destroyed protection group or snapshot. Optional.
addhgrouplist	list	Adds a list of host groups to the existing list. Optional.
addhostlist	list	Adds a list of hosts to the existing list. Optional.
addtargetlist	list	Adds a list of target arrays to the existing list. Optional.
addvollist	list	Adds a list of volumes to the existing list. Optional.
all_for	number	Modifies the retention policy of the protection group. Specifies the length of time to keep the snapshots on the source array before they are eradicated. Optional.
allowed	boolean	Allows (`true`) or disallows (`false`) a protection group from being replicated. Optional.
days	number	Modifies the retention policy of the protection group. Specifies the number of days to keep the per_day snapshots beyond the all_for period before they are eradicated. Optional.
hgrouplist	list	Replaces the list of member host groups. Optional.
hostlist	list	Replaces the list of member hosts. Optional.
name	string	Renames a protection group. Optional.
per_day	number	Modifies the retention policy of the protection group. Specifies the number of per_day snapshots to keep beyond the all_for period. Optional.
remhgrouplist	list	Removes a list of host groups from the existing list. Optional.
remhostlist	list	Removes a list of hosts from the existing list. Optional.

remtargetlist	list	Removes a list of target arrays from the existing list. Optional.
remvollist	list	Removes a list of volumes from the existing list. Optional.
replicate_at	number	Modifies the replication schedule of the protection group. Specifies the preferred time, on the hour, at which to replicate the snapshots. Optional.
replicate_blackout	object	Modifies the replication schedule of the protection group. Specifies the range of time at which to suspend replication. See the example for the dictionary format. Optional.
replicate_enabled	boolean	Used to enable (`true`) or disable (`false`) the protection group replication schedule. Optional.
replicate_frequency	number	Modifies the replication schedule of the protection group. Specifies the replication frequency. Optional.
snap_at	number	Modifies the snapshot schedule of the protection group. Specifies the preferred time, on the hour, at which to generate the snapshot. Optional.
snap_enabled	boolean	Used to enable (`true`) or disable (`false`) the protection group snapshot schedule. Optional.
snap_frequency	number	Modifies the snapshot schedule of the protection group. Specifies the snapshot frequency. Optional.
target_all_for	number	Modifies the retention policy of the protection group. Specifies the length of time to keep the replicated snapshots on the target arrays. Optional.
target_days	number	Modifies the retention policy of the protection group. Specifies the number of days to keep the target_per_day replicated snapshots beyond the target_all_for period before they are eradicated. Optional.
target_per_day	number	Modifies the retention policy of the protection group. Specifies the number of per_day replicated snapshots to keep beyond the target_all_for period. Optional.
targetlist	list	Replaces the list of member target arrays. Optional.
vollist	list	Replaces the list of member volumes. Optional.

Example 1: Rename protection group

Request:

```
PUT https://pure01.example.com/api/1.13/pgroup/pg2
{
  "name": "pg6"
}
```

Response:

```
{
  "name": "pg6"
}
```

Example 2: Set host members

Request:

```
PUT https://pure01.example.com/api/1.13/pgroup/pg4
{
  "hostlist": [
    "h3"
  ]
}
```

Response:

```
{
  "hgroups": null,
  "hosts": [
    "h3"
  ],
  "name": "pg4",
  "source": "pure01",
  "targets": null,
  "volumes": null
}
```

Example 3: Enables the replication schedule for pg4

Request:

```
PUT https://pure01.example.com/api/1.13/pgroup/pg4
{
  "replicate_enabled": true
}
```

Response:

```
{
  "name": "pg4",
  "replicate_at": null,
  "replicate_blackout": null,
  "replicate_enabled": true,
  "replicate_frequency": 14400,
  "snap_at": null,
  "snap_enabled": false,
  "snap_frequency": 3600
}
```

Example 4: Sets a replication blackout window from 2am to 4am (in seconds) for pg4

Request:

```
PUT https://pure01.example.com/api/1.13/pgroup/pg4
{
  "replicate_blackout": {
    "end": 14400,
    "start": 7200
  }
}
```

Response:

```
{
  "name": "pg4",
  "replicate_at": null,
  "replicate_blackout": [
    {
      "end": 14400,
```

```
      "start": 7200
    }
  ],
  "replicate_enabled": true,
  "replicate_frequency": 14400,
  "snap_at": null,
  "snap_enabled": false,
  "snap_frequency": 3600
}
```

Example 5: Sets a retention policy to retain all snapshots for 3 days (in minutes), and keep 12 snapshots per day after the all_for period ends

Request:

```
PUT https://pure01.example.com/api/1.13/pgroup/pg4
{
  "all_for": 4320,
  "per_day": 12
}
```

Response:

```
{
  "all_for": 4320,
  "days": 7,
  "name": "pg4",
  "per_day": 12,
  "target_all_for": 86400,
  "target_days": 7,
  "target_per_day": 4
}
```

6.6 DELETE pgroup/{pgroup}

Destroys or eradicates the specified protection group or protection group snapshot.

Parameters

Parameter	Type	Description
eradicate	boolean	Eradicates the destroyed protection group or protection group snapshot. If destroying or eradicating a protection group, its protection group snapshots are also destroyed or eradicated. Optional.

Example 1: Destroys protection group pg5

Request:

```
DELETE https://pure01.example.com/api/1.13/pgroup/pg5
```

Response:

```
{
  "name": "pg5"
}
```

Example 2: Eradicates protection group pg5

Request:

```
DELETE https://pure01.example.com/api/1.13/pgroup/pg5
{
  "eradicate": true
}
```

Response:

```
{
  "name": "pg5"
}
```

7. Connection Ports

View array host connection port attributes.

7.1 GET port

Lists array ports and the worldwide names assigned to each port.

Parameters

Parameter	Type	Description
initiators	boolean	Displays host worldwide names (both those discovered by Purity and those assigned by administrators) and the array ports (targets) on which they are eligible to communicate. Optional.

Example 1: List all ports

Request:

```
GET https://pure01.example.com/api/1.13/port
```

Response:

```
[
  {
    "failover": null,
    "iqn": null,
    "name": "CT0.FC0",
    "portal": null,
    "wwn": "5001500150015000"
  },
  {
    "failover": null,
    "iqn": null,
    "name": "CT0.FC1",
    "portal": null,
    "wwn": "5001500150015001"
  },
  {
    "failover": null,
    "iqn": null,
    "name": "CT0.FC2",
    "portal": null,
    "wwn": "5001500150015002"
  },
  {
    "failover": null,
    "iqn": null,
    "name": "CT0.FC3",
    "portal": null,
    "wwn": "5001500150015003"
  }
]
```

Example 2: List all initiator ports

Request:

```
GET https://pure01.example.com/api/1.13/port?initiators=true
```

Response:

```
[
```

```
  {
    "failover": null,
    "iqn": null,
    "portal": null,
    "target": null,
    "target_iqn": null,
    "target_portal": null,
    "target_wwn": null,
    "wwn": "0000111122223333"
  },
  {
    "failover": null,
    "iqn": null,
    "portal": null,
    "target": "CT0.FC1",
    "target_iqn": null,
    "target_portal": null,
    "target_wwn": "5001500150015001",
    "wwn": "0008000800080008"
  },
  {
    "failover": null,
    "iqn": null,
    "portal": null,
...
    "target_wwn": "5001500150015003",
    "wwn": "0000000000000000"
  }
]
```

8. Alerts and Messages

View a list of email addresses designated to receive Purity alerts, test the alert feature, enable or disable alert notifications, designate email addresses to receive Purity alerts, and remove email addresses from the list of designated alert recipients. View a list of alert events, audit records, and user login sessions. Flag alert and audit messages.

8.1 GET alert

Lists email recipients that are designated to receive Purity alert messages.

Parameters

Parameter	Type	Description
None	n/a	n/a

Example 1: List all alert recipients

Request:

```
GET https://pure01.example.com/api/1.13/alert
```

Response:

```
[
  {
    "enabled": false,
    "name": "flasharray-alerts@purestorage.com"
  }
]
```

8.2 GET alert/{address}

Lists information about the specified email recipient.

Parameters

Parameter	Type	Description
None	n/a	n/a

Example 1: List a single alert recipient

Request:

```
GET https://pure01.example.com/api/1.13/alert/flasharray-alerts@purestorage.com
```

Response:

```
{
  "enabled": false,
  "name": "flasharray-alerts@purestorage.com"
}
```

8.3 POST alert/{address}

Designates any valid email address to receive Purity alert messages.

Up to 20 addresses can be designated in an array (19 in addition to the built-in `flasharray-alerts@purestorage.com` address.)

Parameters

Parameter	Type	Description
None	n/a	n/a

Example 1: Add a new alert recipient

Request:

```
POST https://pure01.example.com/api/1.13/alert/admin@example.com
```

Response:

```
{
  "enabled": true,
  "name": "admin@example.com"
}
```

8.4 PUT alert

Tests the ability of the array to send alert messages to all of the designated email addresses.

Parameters

Parameter	Type	Description
action	string	Set the parameter to `"test"` to test the ability of the array to send alert messages to all of the designated email addresses. Verification of successful test message transmission is done at the destination. Required.

8.5 PUT alert/{address}

Tests, enables, and disables the transmission of alert messages to the specified email address.

At least one parameter must be specified.

Parameters

Parameter	Type	Description

action	string	Set the parameter to `"test"` to test the ability of the array to send alert messages to the specified email address, designated or not. Verification of successful test message transmission is done at the destination. Optional.
enabled	boolean	Enables (`true`) or disables (`false`) the transmission of alert messages to the specified email address. Optional.

Example 1: Disable a single alert recipient

Request:

```
PUT https://pure01.example.com/api/1.13/alert/admin@example.com
{
  "enabled": false
}
```

Response:

```
{
  "enabled": false,
  "name": "admin@example.com"
}
```

8.6 DELETE alert/{address}

Deletes an email address from the list of addresses designated to receive Purity alert messages. You cannot delete the built-in `flasharray-alerts@purestorage.com` address.

Parameters

Parameter	Type	Description
None	n/a	n/a

Example 1: Delete a single alert recipient

Request:

```
DELETE https://pure01.example.com/api/1.13/alert/admin@example.com
```

Response:

```
{
  "name": "admin@example.com"
}
```

8.7 GET message

Lists alert events, audit records, and user login sessions.

Parameters

Parameter	Type	Description
audit	boolean	Lists audit records instead of alerts (default). May not be used with `login` or `open`. Optional.
flagged	boolean	Lists flagged messages only. The array automatically flags warnings and critical alerts. Optional.

login	boolean	Lists user session logs instead of alerts (default). May not be used with `audit` or `open`. Optional.
open	boolean	Lists open messages. May not be used with `audit` or `login`. Optional.
recent	boolean	Lists recent messages. An audit record is considered recent if it relates to a command issued within the past 24 hours. An alert is considered recent if the situation that triggered it is unresolved, or has only been resolved within the past 24 hours. A user session log event is considered recent if the login, logout, or authentication event occurred within the past 24 hours. Optional.
user	string	When `audit` is set to `true`, `user` can be used to list audit records for a specific user. When `login` is set to `true`, `user` can be used to list user session logs for a specific user. Optional.

Example 1: List all alert messages

Request:

```
GET https://pure01.example.com/api/1.13/message
```

Response:

```
[]
```

Example 2: List all audit trail messages

Request:

```
GET https://pure01.example.com/api/1.13/message?audit=true
```

Response:

```
[
  {
    "component_name": null,
    "component_type": "purearray",
    "details": "--senderdomain example.com",
    "event": "setattr",
    "id": 3,
    "opened": "2017-10-06T21:57:07Z",
    "user": "root"
  },
  {
    "component_name": "pureuser",
    "component_type": "pureadmin",
    "details": "--api-token",
    "event": "create",
    "id": 4,
    "opened": "2017-10-06T21:57:08Z",
    "user": "pureuser"
  },
  {
    "component_name": null,
    "component_type": "puredns",
    "details": "--domain example.com --nameservers 192.168.0.1,192.168.1.1",
    "event": "setattr",
    "id": 5,
...
    "opened": "2017-10-06T22:00:46Z",
    "user": "os76"
  }
]
```

8.8 PUT message/{id}

Flags or unflags a message.

Parameters

Parameter	Type	Description
flagged	boolean	Flags (`true`) or unflags (`false`) a message. If set to `true`, flags the message with the specified ID. If set to `false`, unflags the message. Required.

9. SNMP Manager Connections

View and set SNMP manager and trap message attributes.

9.1 GET snmp

Lists designated SNMP managers and their communication and security attributes.

Parameters

Parameter	Type	Description
engine_id	boolean	SNMP v3 only. If set to `true`, displays the SNMP v3 engine ID generated by Purity for the array. Optional.

Example 1: List all SNMP managers

Request:

```
GET https://pure01.example.com/api/1.13/snmp
```

Response:

```
[
  {
    "auth_passphrase": null,
    "auth_protocol": null,
    "community": null,
    "host": "localhost",
    "name": "localhost",
    "privacy_passphrase": null,
    "privacy_protocol": null,
    "user": null,
    "version": "v2c"
  }
]
```

Example 2: List SNMP v3 engine ID

Request:

```
GET https://pure01.example.com/api/1.13/snmp?engine_id=true
```

Response:

```
{
  "engine_id": "80009e22053915caf6c5bc180da8437a"
}
```

9.2 GET snmp/{manager}

Lists communication and security attributes for the specified SNMP manager.

Parameters

Parameter	Type	Description
None	n/a	n/a

9.3 POST snmp/{manager}

Creates a Purity SNMP manager object that identifies a host (SNMP manager) and specifies the protocol attributes for communicating with it.

Once a manager object is created, the transmission of SNMP traps is immediately enabled.

Parameters

Parameter	Type	Description
auth_passphrase	string	SNMP v3 only. Passphrase used by Purity to authenticate the array with the specified managers. The value must be between 1 and 32 characters in length and from the set {[A-Z], [a-z], [0-9], _ (underscore), and - (hyphen)}. Required if the `auth-protocol` value is set.
auth_protocol	string	SNMP v3 only. Hash algorithm used to validate the authentication passphrase. Valid values are `MD5` or `SHA`. Optional.
community	string	SNMP v2c only. Manager community ID under which Purity is to communicate with the specified managers. The value must be between 1 and 32 characters in length and from the set {[A-Z], [a-z], [0-9], _ (underscore), and - (hyphen)}. Required if the `version` parameter is set to v2c.
host	string	DNS hostname or IP address of a computer that hosts an SNMP manager to which Purity is to send trap messages when it generates alerts. If specifying an IP address, enter the IPv4 or IPv6 address. For IPv4, specify the IP address in the form `ddd.ddd.ddd.ddd`, where ddd is a number ranging from 0 to 255 representing a group of 8 bits. If a port number is also specified, append it to the end of the address in the format `ddd.ddd.ddd.ddd:PORT`, where `PORT` represents the port number. For IPv6, specify the IP address in the form `xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx`, where `xxxx` is a hexadecimal number representing a group of 16 bits. Consecutive fields of zeros can be shortened by replacing the zeros with a double colon (`::`). If a port number is also specified, enclose the entire address in square brackets (`[]`) and append the port number to the end of the address. For example, `[xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx]:PORT`, where `PORT` represents the port number. Required.
privacy_passphrase	string	SNMP v3 only. Passphrase used to encrypt SNMP messages. The value must be between 8 and 63 non-space ASCII characters in length. Required if the `privacy-protocol` value is set.
privacy_protocol	string	SNMP v3 only. Encryption protocol for SNMP messages. Valid values are `AES` or `DES`. Optional.
user	string	SNMP v3 only. User ID recognized by the specified SNMP managers which Purity is to use in communications with them. The value must be between 1 and 32 characters in length and from the set {[A-Z], [a-z], [0-9], _ (underscore), and -(hyphen)}. Optional.
version	string	Version of the SNMP protocol to be used by Purity in communications with the specified manager(s). The `version` value is case sensitive. Valid values are `v2c` (default) and `v3`.

		Optional.

9.4 PUT snmp/{manager}

Changes name or the protocol attributes of the specified SNMP manager.

At least one parameter must be specified.

Parameters

Parameter	Type	Description
action	string	Set the parameter to `"test"` to send a test trap to the specified SNMP manager. Optional.
auth_passphrase	string	SNMP v3 only. Passphrase used by Purity to authenticate the array with the specified managers. The value must be between 1 and 32 characters in length and from the set {[A-Z], [a-z], [0-9], _ (underscore), and - (hyphen)}. Required if the `auth-protocol` value is set.
auth_protocol	string	SNMP v3 only. Hash algorithm used to validate the authentication passphrase. Valid values are `MD5` or `SHA`. Optional.
community	string	SNMP v2c only. Manager community ID under which Purity is to communicate with the specified managers. The value must be between 1 and 32 characters in length and from the set {[A-Z], [a-z], [0-9], _ (underscore), and - (hyphen)}. Required if the `version` parameter is set to v2c.
host	string	DNS hostname or IP address of a computer that hosts an SNMP manager to which Purity is to send trap messages when it generates alerts. If specifying an IP address, enter the IPv4 or IPv6 address. For IPv4, specify the IP address in the form `ddd.ddd.ddd.ddd`, where ddd is a number ranging from 0 to 255 representing a group of 8 bits. If a port number is also specified, append it to the end of the address in the format `ddd.ddd.ddd.ddd:PORT`, where `PORT` represents the port number. For IPv6, specify the IP address in the form `xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx`, where `xxxx` is a hexadecimal number representing a group of 16 bits. Consecutive fields of zeros can be shortened by replacing the zeros with a double colon (`::`). If a port number is also specified, enclose the entire address in square brackets (`[]`) and append the port number to the end of the address. For example, `[xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx]:PORT`, where `PORT` represents the port number. Optional.
name	string	New name of the SNMP manager. SNMP manager names are used in Purity administrative commands, and have no external significance. Changing the name of the SNMP manager does not change the protocol and security attributes of the manager object. Optional.
privacy_passphrase	string	SNMP v3 only. Passphrase used to encrypt SNMP messages. The value must be between 8 and 63 non-space ASCII characters in length. Required if the `privacy_protocol` value is set.
privacy_protocol	string	SNMP v3 only. Encryption protocol for SNMP messages. Valid values are `AES` or `DES`. Optional.
user	string	SNMP v3 only. User ID recognized by the specified SNMP managers which Purity is to use in communications with them. The value must be between 1 and 32 characters in length and from the set {[A-Z], [a-z], [0-9], _ (underscore), and -(hyphen)}.

		Optional.
version	string	Version of the SNMP protocol to be used by Purity in communications with the specified manager(s). The `version` value is case sensitive. Valid values are `v2c` (default) and `v3`. Optional.

9.5 DELETE snmp/{manager}

Stops communication with the specified managers and deletes the SNMP manager object.

Parameters

Parameter	Type	Description
None	n/a	n/a

10. SSL Certificate

View and manage SSL certificates.

10.1 GET cert

Lists all available certificates.

The REST API call returns the attributes of all certificates.

Parameters

Parameter	Type	Description
None	n/a	n/a

10.2 GET cert/{certificate}

Lists attributes or exports the specified certificate.

If the request does not include parameters, the REST API call returns the attributes of the certificate. Include parameters to export the respective certificates. Note: the name 'management' refers to the single certificate which this command managed in versions 1.11 and older.

Parameters

Parameter	Type	Description
ca_certificate	boolean	If set to `true`, exports the current CA certificate. Optional.
certificate	boolean	If set to `true`, exports the current certificate. Optional.
intermediate_certificate	boolean	If set to `true`, exports the current intermediate certificate. Optional.

10.3 GET cert/certificate_signing_request/{certificate}

Constructs a certificate signing request (CSR) for signing by a certificate authority (CA).

Parameters

Parameter	Type	Description
common_name	string	Sets the common name of the CSR. The common name must be the fully qualified domain name of the current array. For example, for https://purearray.example.com, the common name must be purearray.example.com or *.example.com for a wildcard certificate. It can also be the management IP address of the array or the short name of the current array. It can't have more than 64 characters. Optional.
country	string	Sets the country name of the CSR. The country name represents the two-letter ISO code for the country where your organization is located. Optional.
email	string	Sets the email address used to contact your organization. Optional.
locality	string	Sets the name of the city where your organization is located. Optional.
organization	string	Sets the name of your organization. Enter the full and exact legal name of your organization. The organization name should not be abbreviated and should include suffixes such as Inc, Corp, or LLC. For example,'Pure Storage, Inc.' Optional.
organizational_unit	string	Sets the name of the department within your organization that you want to appear in the certificate. Optional.
state	string	Sets the name of the state or province where your organization is located. Optional.

10.4 PUT cert/{certificate}

Creates a self-signed certificate or imports a certificate signed by a certificate authority (CA).

Creating a self-signed certificate or importing a certificate replaces the existing certificate. Either the `certificate` or `self-signed` parameter must be specified.

Parameters

Parameter	Type	Description
ca_certificate	string	Used with the `certificate` parameter to specify the CA certificate to import. Optional.
certificate	string	Imports a certificate signed by a certificate authority. Optional.
common_name	string	Used with the `self-signed` parameter to specify the common name. The common name represents the qualified domain name of the current array. For example, for https://purearray.example.com, the common name must be purearray.example.com or *.example.com for a wildcard certificate. It can also be the management IP address of the array or the short name of the current array. It can't have more than 64 characters. Optional.
country	string	Used with the `self-signed` parameter to specify the country name. The country name represents the two-letter ISO code for the country where your organization is located. Optional.
days	int	`self-signed`

		Used with the parameter to specify the number of valid days for the self-signed certificate being generated. Optional.
email	string	Used with the `self-signed` parameter to specify the email address used to contact your organization. Optional.
intermediate_certificate	string	Used with the `certificate` parameter to specify the intermediate certificate to import. Optional.
key	string	Used with the `certificate` parameter to specify the private key to import. Optional.
key-size	int	Used with the `new_key` parameter to specify the key size in bits. Valid values are 512, 1024, 2048 (default) or 4096. A key size smaller than 2048 is considered insecure. Optional.
locality	string	Used with the `self-signed` parameter to specify the city where your organization is located. Optional.
new_key	boolean	Used with the `self-signed` parameter to create a new key when creating the self-signed certificate. If a new private key is not generated, the certificate uses the existing private key. Optional.
organization	string	Used with the `self-signed` parameter to specify the full and exact legal name of your organization. The organization name should not be abbreviated and should include suffixes such as Inc, Corp, or LLC. For example,'Pure Storage, Inc.' Optional.
organizational_unit	string	Used with the `self-signed` parameter to specify the department within your organization that you want to appear in the certificate. Optional.
passphrase	string	Used with the `key` parameter to decrypt the private key. Optional.
self_signed	boolean	Generates a self-signed certificate. `self_signed` or `certificate` must be set. Optional.
state	string	Used with the `self-signed` parameter to specify the state or province where your organization is located. Optional.

10.5 POST cert/{certificate}

Creates (and optionally initializes) a new certificate.

Creating a new self-signed certificate or importing a certificate does not affect existing certificates, and there may not already be a certificate with the same name. Either the `certificate` or `self-signed` parameter must be specified.

Parameters

Parameter	Type	Description
ca_certificate	string	Used with the `certificate` parameter to specify the CA certificate to import. Optional.

certificate	string	Imports a certificate signed by a certificate authority. Optional.
common_name	string	Used with the `self-signed` parameter to specify the common name. The common name represents the qualified domain name of the current array. For example, for https://purearray.example.com, the common name must be purearray.example.com or *.example.com for a wildcard certificate. It can also be the management IP address of the array or the short name of the current array. It can't have more than 64 characters. Optional.
country	string	Used with the `self-signed` parameter to specify the country name. The country name represents the two-letter ISO code for the country where your organization is located. Optional.
days	int	Used with the `self-signed` parameter to specify the number of valid days for the self-signed certificate being generated. Optional.
email	string	Used with the `self-signed` parameter to specify the email address used to contact your organization. Optional.
intermediate_certificate	string	Used with the `certificate` parameter to specify the intermediate certificate to import. Optional.
key	string	Used with the `certificate` parameter to specify the private key to import. Optional.
key-size	int	Used with the `new_key` parameter to specify the key size in bits. Valid values are 512, 1024, 2048 (default) or 4096. A key size smaller than 2048 is considered insecure. Optional.
locality	string	Used with the `self-signed` parameter to specify the city where your organization is located. Optional.
new_key	boolean	Used with the `self-signed` parameter to create a new key when creating the self-signed certificate. If a new private key is not generated, the certificate uses the existing private key. Optional.
organization	string	Used with the `self-signed` parameter to specify the full and exact legal name of your organization. The organization name should not be abbreviated and should include suffixes such as Inc, Corp, or LLC. For example,'Pure Storage, Inc.' Optional.
organizational_unit	string	Used with the `self-signed` parameter to specify the department within your organization that you want to appear in the certificate. Optional.
passphrase	string	Used with the `key` parameter to decrypt the private key. Optional.
self_signed	boolean	Generates a self-signed certificate. `self_signed` or `certificate` must be set. Optional.
state	string	Used with the `self-signed` parameter to specify the state or province where your organization is located.

		Optional.

10.6 DELETE cert/{certificate}

Delete a certificate. The 'management' certificate may not be deleted.

The REST API call returns the name of the certificate that was deleted.

Parameters

Parameter	Type	Description
None	n/a	n/a

11. Network Interface

View and set DNS and network interface attributes.

11.1 GET dns

Lists DNS attributes for the array administrative network.

Parameters

Parameter	Type	Description
None	n/a	n/a

Example 1: List DNS settings

Request:

```
GET https://pure01.example.com/api/1.13/dns
```

Response:

```
{
  "domain": "example.com",
  "nameservers": [
    "192.168.0.1",
    "192.168.1.1"
  ]
}
```

11.2 PUT dns

Sets the DNS attributes for the array administrative network.

At least one parameter must be specified.

Parameters

Parameter	Type	Description
domain	string	Domain suffix to be appended by the array when performing DNS lookups. Optional.
nameservers	list	A list of up to three DNS server IP addresses that replace the current list of name servers. For IPv4, specify the IP address in the form `ddd.ddd.ddd.ddd`, where `ddd` is a number ranging from 0 to 255 representing a group of 8 bits. For IPv6, specify the IP address in the form `xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx`, where `xxxx` is a hexadecimal number representing a group of 16 bits. When specifying an IPv6 address,

		consecutive fields of zeros can be shortened by replacing the zeros with a double colon (`::`). Optional.

Example 1: Modify DNS settings

Request:

```
PUT https://pure01.example.com/api/1.13/dns
{
  "domain": "example.com",
  "nameservers": [
    "192.168.1.1",
    "192.168.0.1"
  ]
}
```

Response:

```
{
  "domain": "example.com",
  "nameservers": [
    "192.168.0.1",
    "192.168.1.1"
  ]
}
```

11.3 GET subnet

Lists subnets and their statuses (enabled or disabled) and attributes.

Parameters

Parameter	Type	Description
None	n/a	n/a

Example 1: List subnets

Request:

```
GET https://pure01.example.com/api/1.13/subnet
```

Response:

```
[
  {
    "enabled": true,
    "gateway": null,
    "interfaces": [],
    "mtu": 1500,
    "name": "subnet100",
    "prefix": "192.168.0.0/24",
    "services": [],
    "vlan": 100
  }
]
```

11.4 GET subnet/{subnet}

Lists the attributes of the specified subnet.

Parameters

Parameter	Type	Description
None	n/a	n/a

Example 1: List settings for subnet subnet100

Request:

```
GET https://pure01.example.com/api/1.13/subnet/subnet100
```

Response:

```
{
  "enabled": true,
  "gateway": null,
  "interfaces": [],
  "mtu": 1500,
  "name": "subnet100",
  "prefix": "192.168.0.0/24",
  "services": [],
  "vlan": 100
}
```

11.5 POST subnet/{subnet}

Creates a subnet with the specified name.

At least one parameter must be specified.

Parameters

Parameter	Type	Description
gateway	string	Sets the IP address of the gateway through which the specified subnet is to communicate with the network. For IPv4, specify the gateway IP address in the form `ddd.ddd.ddd.ddd`. For IPv6, specify the gateway IP address in the form `xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx/xxx`. When specifying an IPv6 address, consecutive fields of zeros can be shortened by replacing the zeros with a double colon (`::`). Optional.
mtu	number	Maximum message transfer unit (packet) size for the subnet in bytes. Valid values are integers between 1280 and 9216 (inclusive). The default value is 1500. Optional.
prefix	string	Sets the IP address to be associated with the specified subnet. Specify the the subnet prefix and prefix length. For IPv4, specify the subnet prefix in the form `ddd.ddd.ddd.ddd/dd`. For IPv6, specify the subnet prefix in the form `xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx/xxx`. When specifying an IPv6 address, consecutive fields of zeros can be shortened by replacing the zeros with a double colon (`::`). Optional.
vlan	number	VLAN ID. Optional.

Example 1: Create subnet mgmt with the specified gateway address and prefix

Request:

```
POST https://pure01.example.com/api/1.13/subnet/mgmt
{
  "gateway": "192.168.100.1",
  "prefix": "192.168.100.0/24"
}
```

Response:

```
{
  "enabled": true,
```

```
  "gateway": "192.168.100.1",
  "interfaces": null,
  "mtu": 1500,
  "name": "mgmt",
  "prefix": "192.168.100.0/24",
  "services": null,
  "vlan": null
}
```

11.6 PUT subnet/{subnet}

Modifies the attributes of the subnet.

At least one parameter must be specified.

Parameters

Parameter	Type	Description
enabled	boolean	Enables or disables a subnet. Optional. Cannot be used with any other parameters.
gateway	string	Sets the IP address of the gateway through which the specified subnet is to communicate with the network. For IPv4, specify the gateway IP address in the form `ddd.ddd.ddd.ddd`. For IPv6, specify the gateway IP address in the form `xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx/xxx`. When specifying an IPv6 address, consecutive fields of zeros can be shortened by replacing the zeros with a double colon (`::`). Optional.
mtu	number	Maximum message transfer unit (packet) size for the subnet in bytes. Valid values are integers between 1280 and 9216 (inclusive). The default value is 1500. Optional.
name	string	Renames the specified subnet. Optional. Cannot be used with any other parameters.
prefix	string	Sets the IP address to be associated with the specified subnet. Specify the the subnet prefix and prefix length. For IPv4, specify the subnet prefix in the form `ddd.ddd.ddd.ddd/dd`. For IPv6, specify the subnet prefix in the form `xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx/xxx`. When specifying an IPv6 address, consecutive fields of zeros can be shortened by replacing the zeros with a double colon (`::`). Optional.
vlan	number	VLAN ID. Optional.

Example 1: Change VLAN ID for subnet mgmt

Request:

```
PUT https://pure01.example.com/api/1.13/subnet/mgmt
{
  "vlan": 100
}
```

Response:

```
{
  "enabled": true,
  "gateway": "192.168.100.1",
  "interfaces": [],
  "mtu": 1500,
  "name": "mgmt",
  "prefix": "192.168.100.0/24",
  "services": [],
  "vlan": 100
}
```

Example 2: Clear gateway for subnet mgmt

Request:

```
PUT https://pure01.example.com/api/1.13/subnet/mgmt
{
  "gateway": null
}
```

Response:

```
{
  "enabled": true,
  "gateway": null,
  "interfaces": [],
  "mtu": 1500,
  "name": "mgmt",
  "prefix": "192.168.100.0/24",
  "services": [],
  "vlan": 100
}
```

Example 3: Disable subnet mgmt

Request:

```
PUT https://pure01.example.com/api/1.13/subnet/mgmt
{
  "enabled": false
}
```

Response:

```
{
  "enabled": false,
  "gateway": null,
  "interfaces": [],
  "mtu": 1500,
  "name": "mgmt",
  "prefix": "192.168.100.0/24",
  "services": [],
  "vlan": 100
}
```

Example 4: Rename subnet mgmt

Request:

```
PUT https://pure01.example.com/api/1.13/subnet/mgmt
{
  "name": "managementSubnet"
}
```

Response:

```
{
  "enabled": false,
  "gateway": null,
  "interfaces": [],
  "mtu": 1500,
  "name": "managementSubnet",
  "prefix": "192.168.100.0/24",
  "services": [],
  "vlan": 100
}
```

11.7 DELETE subnet/{subnet}

Deletes the specified subnet.

Parameters

Parameter	Type	Description
None	n/a	n/a

Example 1: Delete subnet managementSubnet

Request:

```
DELETE https://pure01.example.com/api/1.13/subnet/managementSubnet
```

Response:

```
{
  "name": "managementSubnet"
}
```

11.8 GET network

Lists array administrative network interfaces and their statuses (enabled or disabled) and attributes.

Parameters

Parameter	Type	Description
None	n/a	n/a

Example 1: List network interfaces

Request:

```
GET https://pure01.example.com/api/1.13/network
```

Response:

```
[
  {
    "address": "10.14.225.210",
    "enabled": true,
    "gateway": "10.14.224.1",
    "hwaddr": "00:50:56:a5:98:08",
    "mtu": 1500,
    "name": "ct0.eth0",
    "netmask": "255.255.240.0",
    "services": [
      "management",
      "replication"
    ],
    "slaves": [],
    "speed": 1000000000,
    "subnet": null
  },
  {
    "address": null,
    "enabled": true,
    "gateway": null,
    "hwaddr": "00:50:56:a5:2b:6d",
    "mtu": 1500,
    "name": "ct0.eth1",
    "netmask": null,
...
    "speed": 10000000000,
    "subnet": null
  }
]
```

11.9 GET network/{network_component}

Lists the attributes for the specified network component.

Parameters

Parameter	Type	Description
None	n/a	n/a

Example 1: List settings for a single network interface

Request:

```
GET https://pure01.example.com/api/1.13/network/ct0.eth0
```

Response:

```
{
  "address": "10.14.225.210",
  "enabled": true,
  "gateway": "10.14.224.1",
  "hwaddr": "00:50:56:a5:98:08",
  "mtu": 1500,
  "name": "ct0.eth0",
  "netmask": "255.255.240.0",
  "services": [
    "management",
    "replication"
  ],
  "slaves": [],
  "speed": 10000000000,
  "subnet": null
}
```

11.10 POST network/vif/{vlan_interface}

Creates a VLAN interface.

Parameters

Parameter	Type	Description
address	string	Sets the IP address to be associated with the specified interface. The IP address must match the subnet prefix. For IPv4, enter the address in CIDR notation `ddd.ddd.ddd.ddd/dd`. For example, `10.20.20.210/24`. Alternatively, specify the address `ddd.ddd.ddd.ddd` with a netmask. The netmask is set through the `netmask` parameter. For IPv6, enter the address and prefix length in the form `xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx/xxx`. For example, `2001:0db8:85a3::ae26:8a2e:0370:7334/64`. Consecutive fields of zeros can be shortened by replacing the zeros with a double colon (`::`). Alternatively, specify the address `xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx` with a prefix length. The prefix length is set through the `netmask` parameter. Optional.
subnet	string	Sets the subnet associated with the specified interface. Required.

Example 1: Create a VLAN interface and associate it with subnet subnet100.

Request:

```
POST https://pure01.example.com/api/1.13/network/vif/ct0.eth1.100
{
```

```
  "address": "192.168.0.100",
  "subnet": "subnet100"
}
```

Response:

```
{
  "address": "192.168.0.100",
  "enabled": true,
  "gateway": null,
  "hwaddr": "00:50:56:a5:2b:6d",
  "mtu": 1500,
  "name": "ct0.eth1.100",
  "netmask": "255.255.255.0",
  "services": [
    "iscsi",
    "management"
  ],
  "slaves": [],
  "speed": 100000000000,
  "subnet": "subnet100"
}
```

11.11 PUT network/{network_component}

Performs network interface actions and sets the administrative network component attributes.

At least one parameter must be specified.

Parameters

Parameter	Type	Description
address	string	Sets the IP address to be associated with the specified Ethernet interface. For IPv4, enter the address in CIDR notation `ddd.ddd.ddd.ddd/dd`. For example, `10.20.20.210/24`. Alternatively, specify the address `ddd.ddd.ddd.ddd` with a netmask. The netmask is set through the `netmask` parameter. For IPv6, enter the address and prefix length in the form `xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx/xxx`. For example, `2001:0db8:85a3::ae26:8a2e:0370:7334/64`. Consecutive fields of zeros can be shortened by replacing the zeros with a double colon (`::`). Alternatively, specify the address `xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx` with a prefix length. The prefix length is set through the `netmask` parameter. Required if the `netmask` parameter is set.
addsubinterfacelist	list	Add subinterfaces to the existing list. May only be used on network bond subinterfaces. Optional.
enabled	boolean	Enables (`true`) or disables (`false`) communication between the specified Ethernet interface and the administrative network. Do not disable the interface through which the administrative session is being conducted. Optional.
gateway	string	Sets the IP address of the gateway through which the specified interface is to communicate with the network. For IPv4, specify the gateway IP address in the form `ddd.ddd.ddd.ddd`. For IPv6, specify the gateway IP address in the form `xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx/xxx`. When specifying an IPv6 address, consecutive fields of zeros can be shortened by replacing the zeros with a double colon (`::`). Optional.
mtu	number	Maximum message transfer unit (packet) size for the interface in bytes. Valid values are integers between 1280 and 9216 (inclusive). The default value is 1500. Optional.
netmask	string	Used with the `address` parameter to define the range of IP addresses that make up a group of IP addresses

		on the same network. For IPv4, if the address is not entered in CIDR notation, enter the subnet mask in the form `ddd.ddd.ddd.ddd`. For example, `255.255.255.0`. For IPv6, if the address entered does not include a prefix length, specify the prefix length. For example, `64`. Optional.
remsubinterfacelist	list	Removes the list of subinterfaces from the existing list. May only be used on network bond subinterfaces. Optional.
subnet	string	Associates a subnet with the specified VLAN interface. Optional.
subnterfacelist	list	Replaces the existing subinterface list. May only be used on network bond subinterfaces. Optional.

Example 1: Disable network interface ct0.eth1

Request:

```
PUT https://pure01.example.com/api/1.13/network/ct0.eth1
{
  "enabled": false
}
```

Response:

```
{
  "address": null,
  "enabled": false,
  "gateway": null,
  "hwaddr": "00:50:56:a5:2b:6d",
  "mtu": 1500,
  "name": "ct0.eth1",
  "netmask": null,
  "services": [
    "iscsi",
    "management"
  ],
  "slaves": [],
  "speed": 10000000000,
  "subnet": null
}
```

Example 2: Enable network interface ct0.eth1

Request:

```
PUT https://pure01.example.com/api/1.13/network/ct0.eth1
{
  "enabled": true
}
```

Response:

```
{
  "address": null,
  "enabled": true,
  "gateway": null,
  "hwaddr": "00:50:56:a5:2b:6d",
  "mtu": 1500,
  "name": "ct0.eth1",
  "netmask": null,
  "services": [
    "iscsi",
    "management"
  ],
  "slaves": [],
  "speed": 10000000000,
  "subnet": null
}
```

11.12 DELETE network/{vlan_interface}

Deletes a VLAN interface.

Parameters

Parameter	Type	Description
None	n/a	n/a

Example 1: Delete VLAN network interface ct0.eth1.100.

Request:

```
DELETE https://pure01.example.com/api/1.13/network/ct0.eth1.100
```

Response:

```
{
  "name": "ct0.eth1.100"
}
```

12. Hardware Components

View and manage array hardware components.

12.1 GET hardware

Lists array hardware component information.

Returns information about array hardware components that are capable of reporting their status. The display is primarily useful for diagnosing hardware-related problems.

Parameters

Parameter	Type	Description
None	n/a	n/a

Example 1: List all hardware devices

Request:

```
GET https://pure01.example.com/api/1.13/hardware
```

Response:

```
[
  {
    "details": null,
    "identify": "off",
    "index": 0,
    "name": "CT0",
    "slot": null,
    "speed": null,
    "status": "ok",
    "temperature": null,
    "voltage": null
  },
  {
    "details": null,
    "identify": null,
    "index": 0,
    "name": "CT0.ETH0",
    "slot": null,
    "speed": 1000000000,
    "status": "ok",
```

```
    "temperature": null,
    "voltage": null
  },
  {
    "details": null,
...
    "temperature": 25,
    "voltage": null
  }
]
```

12.2 GET hardware/{component}

Lists the attributes for the specified hardware component.

Parameters

Parameter	Type	Description
None	n/a	n/a

Example 1: List a single hardware device

Request:

```
GET https://pure01.example.com/api/1.13/hardware/SH0.BAY0
```

Response:

```
{
  "details": null,
  "identify": "off",
  "index": 0,
  "name": "SH0.BAY0",
  "slot": null,
  "speed": null,
  "status": "ok",
  "temperature": null,
  "voltage": null
}
```

12.3 PUT hardware/{component}

Controls the visual identification of the specified chassis, controllers, flash module bays, NVRAM module bays, and storage shelves.

At least one parameter must be specified.

Parameters

Parameter	Type	Description
identify	string	Turns the LED light on or off to identify the specified component. Valid for chassis, controllers, flash module bays, NVRAM module bays, and storage shelves. Set identify to `"on"` to turn on the LED and identify the specified component. Set identify to `"off"` to turn off the LED. Optional.
index	string	Displays the specified number on the identifying component. Optional.

Example 1: Turn on ID light for SH0.BAY0

Request:

```
PUT https://pure01.example.com/api/1.13/hardware/SH0.BAY0
{
```

```
  "identify": "on"
}
```

Response:

```
{
  "identify": "on",
  "index": 0,
  "name": "SH0.BAY0",
  "slot": null
}
```

Example 2: Turn off ID light

Request:

```
PUT https://pure01.example.com/api/1.13/hardware/SH0.BAY0
{
  "identify": "off"
}
```

Response:

```
{
  "identify": "off",
  "index": 0,
  "name": "SH0.BAY0",
  "slot": null
}
```

12.4 GET drive

Lists flash modules, NVRAM modules, and their attributes.

Parameters

Parameter	Type	Description
None	n/a	n/a

Example 1: List all flash and NVRAM devices

Request:

```
GET https://pure01.example.com/api/1.13/drive
```

Response:

```
[
  {
    "capacity": 461373440,
    "details": null,
    "last_evac_completed": "1970-01-01T00:00:00Z",
    "last_failure": "1970-01-01T00:00:00Z",
    "name": "SH0.BAY0",
    "protocol": "SAS",
    "status": "healthy",
    "type": "SSD"
  },
  {
    "capacity": 461373440,
    "details": null,
    "last_evac_completed": "1970-01-01T00:00:00Z",
    "last_failure": "1970-01-01T00:00:00Z",
    "name": "SH0.BAY1",
    "protocol": "SAS",
    "status": "healthy",
    "type": "SSD"
  },
  {
    "capacity": 461373440,
```

```
    "details": null,
    "last_evac_completed": "1970-01-01T00:00:00Z",
...
    "status": "healthy",
    "type": "NVRAM"
  }
]
```

12.5 GET drive/{drive}

Lists the attributes for the specified flash module or NVRAM module.

Parameters

Parameter	Type	Description
None	n/a	n/a

Example 1: List a single flash module or NVRAM module.

Request:

```
GET https://pure01.example.com/api/1.13/drive/SH0.BAY0
```

Response:

```
{
  "capacity": 461373440,
  "details": null,
  "last_evac_completed": "1970-01-01T00:00:00Z",
  "last_failure": "1970-01-01T00:00:00Z",
  "name": "SH0.BAY0",
  "protocol": "SAS",
  "status": "healthy",
  "type": "SSD"
}
```

13. Apps

View app attributes.

13.1 GET app

Lists the apps that are installed on the array, along with the attributes of each app.

Parameters

Parameter	Type	Description
None	n/a	n/a

14. Users

View and manage users and directory service configurations.

14.1 GET admin

Lists public key and API token information for all users.

Set at least one parameter to true.

Parameters

Parameter	Type	Description

api_token	boolean	If set to `true`, displays a list of users that have REST API access and the dates in which the API tokens were created. Optional.
expose	boolean	If `api_token` is set to `true`, set the `expose` parameter to `true` to unmask the current user's API token. Does not unmmask other users' tokens. Optional.
publickey	boolean	If set to `true`, displays a list of users that have public key access. Optional.

Example 1: List users who have public keys

Request:

```
GET https://pure01.example.com/api/1.13/admin?publickey=true
```

Response:

```
[]
```

Example 2: List users who have API tokens

Request:

```
GET https://pure01.example.com/api/1.13/admin?api_token=true
```

Response:

```
[
  {
    "api_token": "****",
    "created": "2017-10-06T21:58:34Z",
    "name": "os76"
  },
  {
    "api_token": "****",
    "created": "2017-10-06T21:57:08Z",
    "name": "pureuser"
  }
]
```

14.2 GET admin/{user}

Lists public key information for the specified user.

Parameters

Parameter	Type	Description
publickey	boolean	If set to `true`, displays the public key information for the specified user. Required.

14.3 GET admin/{user}/apitoken

Lists the API token for the specified user.

Parameters

Parameter	Type	Description
None	n/a	n/a

14.4 POST admin/{user}/apitoken

Creates an API token for the specified user.

Parameters

Parameter	Type	Description
None	n/a	n/a

14.5 PUT admin

Clears all user permission cache entries.

User permission cache entries are also automatically updated when the user starts a new session.

Parameters

Parameter	Type	Description
action	string	When set to `refresh`, refreshes all user permission cache entries. Required.
clear	boolean	When set to `true`, clears the entire user permission cache. Required. Used with the `action` parameter.

14.6 PUT admin/{user}

Sets the password or public key or refreshes the user permission cache entries for the specified user.

Either the `publickey`, `password`, or `action` parameter must be specified.

Parameters

Parameter	Type	Description
action	string	When set to `refresh`, refreshes the user permission cache entries for the specified user. Cache entries are also automatically updated when the user starts a new session. Optional.
clear	boolean	If the `action` parameter is set to `"refresh"`, set the `clear` parameter to `true` to clear the user permission cache for the specified user. Optional. Used with the `action` parameter.
old_password	string	Used with the `password` parameter to change the password for the single, local administrative account `pureuser`. Set the parameter to the old password. Required if the `password` parameter is set.
password	string	Used with the `old_password` parameter to change the password for the single, local administrative account `pureuser`. The value must be between 1 and 32 characters in length and be entered from a standard English (U.S.) keyboard. Optional.

publickey	string	Changes the public key for SSH access for the specified user. Only system administrators can change public keys on behalf of other users. If no users are provided as arguments, a request to change the public key will be for the administrator issuing the request and a request to display set public keys will show all users with a public key configured. Optional.

14.7 DELETE admin/{user}/apitoken

Deletes API token for the specified user.

Parameters

Parameter	Type	Description
None	n/a	n/a

14.8 GET directoryservice

Lists current base configuration information for the directory service.

Parameters

Parameter	Type	Description
certificate	boolean	If set to `true`, returns information about the currently configured CA certificate data. Optional.
groups	boolean	If set to `true`, returns information about the group configuration. Optional.

Example 1: List the directory service configuration

Request:

```
GET https://pure01.example.com/api/1.13/directoryservice
```

Response:

```
{
  "base_dn": "DC=ad1,DC=example,DC=com",
  "bind_password": "****",
  "bind_user": "readonlyuser",
  "check_peer": false,
  "enabled": true,
  "uri": [
    "ldaps://ad1.example.com"
  ]
}
```

Example 2: List the current configured CA certificate

Request:

```
GET https://pure01.example.com/api/1.13/directoryservice?certificate=true
```

Response:

```
{
  "certificate": "-----BEGIN CERTIFICATE-----
\nMIIE6jCCBFOgAwIBAgIDEIGKMA0GCSqGSIb3DQEBBQUAME4xCzAJBgNVBAYTAlVT\nMRAwDgYDVQQKEwdFcXVpZmF4MS0wKwYDVQQLEyRFcXVp
ZmF4IFNlY3VyZSBDZXJ0\naWZpY2F0ZSBBdXRob3JpdHkwHhcNMTAwNDAxMjMwMDE0WhcNMTUwNzAzMDQ1MDAw\nWjCBjzEpMCcGA1UEBRMgMmc4
```

```
YU81d0kxYktKMlpENTg4VXNMdkRlM2dUYmc4RFUx\nCzAJBgNVBAYTAlVTMRMwEQYDVQQIEwpDYWxpZm9ybmlhMRIwEAYDVQQHEwlTdW5u\neXZh
bGUxFDASBgNVBAoTC1haG9vICBJbmMuMRYwFAYDVQQDEw13d3cueWFob28u\nY29tMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA6Z
M1jHCkL8rlEKse\n1riTTxyC3WvYQ5m34TlFK7dK4QFI/HPttKGqQm3aVB1Fqi0aiTxe4YQMbd++jnKt\ndjxcpi7sJlFxjMZs4umr1eGo2KgTgS
BAJyhxo23k+VpK1SprdPyM3yEfQVdV7JWC\n4Y71CE2nE6+GbsIuhk/to+jJMO7jXx/430jvo8vhNPL6GvWe/D6ObbnxS72ynLSd\nmLtaltykOv
ZEZiXbbFKgIaYYmCgh89FGVvBkUbGM/Wb5Voiz7ttQLLxKOYRj8Mdk\nTZtzPkM9scIFG1naECPvCxw0NyMyxY3nFOdjUKJ79twanmfCclX2ZO/r
k1CpiOuw\nlrrr/QIDAQABo4ICDjCCAgowDgYDVR0PAQH/BAQDAgTwMB0GA1UdDgQWBBSmrfKs\n68m+dDUSf+S7xJrQ/FXAlzA6BgNVHR8EMzAx
MC+gLaArhilodHRwOi8vY3JsLmdl\nb3RydXN0LmNvbS9jcmxzL3NlY3VyZWNhLmNybDCCAVsGA1UdEQSCAVIwggFOgg13\nd3cueWFob28uY29t
ggl5YWhvby5jb22CDHVzLnlhaG9vLmNvbYIMa3IueWFob28u\nY29tggx1ay55YWhvby5jb22CDGllLnlhaG9vLmNvbYIMZnIueWFob28uY29tgg
xp\nbi55YWhvby5jb22CDGNhLnlhaG9vLmNvbYIMYnIueWFob28uY29tggxkZS55YWhv\nby5jb22CDGVzLnlhaG9vLmNvbYIMbXgueWFob28uY2
9tggxpdC55YWhvby5jb22C\nDHNnLnlhaG9vLmNvbYIMaWQueWFob28uY29tggxwaC55YWhvby5jb22CDHFjLnlh\naG9vLmNvbYIMdHcueWFob2
8uY29tggxoay55YWhvby5jb22CDGNuLnlhaG9vLmNv\nbYIMYXUueWFob28uY29tggxhci55YWhvby5jb22CDHZuLnlhaG9vLmNvbTAfBgNV\nHS
MEGDAWgBRI5mj5K9KylddH2CMgEE8zmJCf1DAdBgNVHSUEFjAUBggrBgEFBQcD\nAQYIKwYBBQUHAwIwDQYJKoZIhvcNAQEFBQADgYEAp9WOMtcD
MM5T0yfPecGv5QhH\nRJZRzgeMPZitLksr1JxxicJrdgv82NWq1bw8aMuRj47ijrtaTEWXaCQCy00yXodD\nzoRJVNoYIvY1arYZf5zv9VZjN5I0
HqUc39mNMe9XdZtbkWE+K6yVh6OimKLbizna\ninu9YTrN/4P/w6KzHho=\n-----END CERTIFICATE-----"
}
```

Example 3: List the directory service group configuration

Request:

```
GET https://pure01.example.com/api/1.13/directoryservice?groups=true
```

Response:

```
{
  "array_admin_group": "pureadmins",
  "group_base": "OU=PureStorage,OU=SAN,OU=IT,OU=US",
  "readonly_group": "purereadonly",
  "storage_admin_group": "pureusers"
}
```

14.9 PUT directoryservice

Modifies and tests the directory service configuration.

At least one parameter must be specified.

Parameters

Parameter	Type	Description
action	string	When set to `test`, tests the current directory service configuration; verifies that the URIs can be resolved and that Purity can bind and query the tree using the bind user credentials. The call also verifies that it can find all the configured groups to ensure the Common Names and group base are correctly configured. Optional.
array_admin_group	string	Sets the common Name (CN) of the directory service group containing administrators with full privileges when managing the FlashArray. The name should be just the Common Name of the group without the CN= specifier. Common Names should not exceed 64 characters in length. Optional.
auto_fetch	boolean	When used with `certificate`, set the value to `true` to download and confirm the server certificate. Optional.
base_dn	string	Sets the base of the Distinguished Name (DN) of the directory service groups. The base should consist of only Domain Components (DCs). The `base_dn` will populate with a default value when a URI is entered by parsing domain components from the URI. The base DN should specify DC= for each domain component and multiple DCs should be separated by commas. Optional.
bind_password	string	Sets the password of the bind_user user name account. Optional.
bind_user	string	Sets the user name that can be used to bind to and query the directory. For Active Directory, enter the username - often referred to as sAMAccountName or User Logon Name - of

		the account that is used to perform directory lookups. For OpenLDAP, enter the full DN of the user. Optional.
certificate	string	Certificate string in PEM format, including the `"-----BEGIN CERTIFICATE-----"` and `"-----END CERTIFICATE-----"` lines. Optional.
check_peer	boolean	Enables (`true`) or disables (`false`) server authenticity enforcement with the configured CA certificate. Optional.
enabled	boolean	Enables (`true`) or disables (`false`) directory service support. Optional.
group_base	string	Specifies where the configured groups are located in the directory tree. This field consists of Organizational Units (OUs) that combine with the base DN attribute and the configured group CNs to complete the full Distinguished Name of the groups. The group base should specify OU= for each OU and multiple OUs should be separated by commas. The order of OUs is important and should get larger in scope from left to right. Each OU should not exceed 64 characters in length. Optional.
readonly_group	string	Sets the common Name (CN) of the configured directory service group containing users with read-only privileges on the FlashArray. This name should be just the Common Name of the group without the CN= specifier. Common Names should not exceed 64 characters in length. Optional.
storage_admin_group	string	Sets the common Name (CN) of the configured directory service group containing administrators with storage-related privileges on the FlashArray. This name should be just the Common Name of the group without the CN= specifier. Common Names should not exceed 64 characters in length. Optional.
trust	boolean	When used with `certificate`, set the value to `true` to skip certificate chain trust verification. Optional.
uri	list	A list of up to 30 URIs of the directory servers. Each URI must include the scheme `ldap://` or `ldaps://` (for LDAP over SSL), a hostname, and a domain name or IP address. For example, `ldap://ad.company.com` configures the directory service with the hostname "ad" in the domain "company.com" while specifying the unencrypted LDAP protocol. If specifying a domain name, it should be resolvable by the configured DNS servers. If specifying an IP address, for IPv4, specify the IP address in the form `ddd.ddd.ddd.ddd`, where `ddd` is a number ranging from 0 to 255 representing a group of 8 bits. For IPv6, specify the IP address in the form `[xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx]`, where `xxxx` is a hexadecimal number representing a group of 16 bits. Enclose the entire address in square brackets (`[]`). Consecutive fields of zeros can be shortened by replacing the zeros with a double colon (`::`). If the scheme of the URIs is `ldaps://`, SSL is enabled. SSL is either enabled or disabled globally, so the scheme of all supplied URIs must be the same. They must also all have the same domain. If base DN is not configured and a URI is provided, the base DN will automatically default to the domain components of the URIs. Optionally specify a port. If a port number is specified, append it to the end of the address. Default ports are 389 for ldap, and 636 for ldaps. Non-standard ports can be specified in the URI if they are in use. Optional.

Example 1: Disable the directory service

Request:

```
PUT https://pure01.example.com/api/1.13/directoryservice
{
  "enabled": false
}
```

Response:

```
{
  "base_dn": "DC=ad1,DC=example,DC=com",
  "bind_password": "****",
  "bind_user": "readonlyuser",
  "check_peer": false,
  "enabled": false,
  "uri": [
    "ldaps://ad1.example.com"
  ]
}
```

Example 2: Enable the directory service

Request:

```
PUT https://pure01.example.com/api/1.13/directoryservice
{
  "enabled": true
}
```

Response:

```
{
  "base_dn": "DC=ad1,DC=example,DC=com",
  "bind_password": "****",
  "bind_user": "readonlyuser",
  "check_peer": false,
  "enabled": true,
  "uri": [
    "ldaps://ad1.example.com"
  ]
}
```