
HDF REST API Documentation

Release 0.1

The HDF Group

Jul 20, 2018

Contents

1	Introduction	3
2	Domains	5
2.1	Mapping of file paths to domain names	5
2.2	Creating Domains	6
2.3	Getting Information about Domains	6
2.4	Deleting Domains	6
2.5	List of Operations	6
3	Groups	13
3.1	Creating Groups	13
3.2	Getting information about Groups	13
3.3	Updating Links	13
3.4	Deleting Groups	14
3.5	List of Operations	14
4	Datasets	41
4.1	Creating Datasets	41
4.2	Getting information about a dataset	41
4.3	Writing data to a dataset	41
4.4	Reading data from a dataset	42
4.5	Resizable datasets	42
4.6	Deleting datasets	42
4.7	List of Operations	42
5	Committed Datatypes	77
5.1	Creating committed datatypes	77
5.2	Getting information about a committed datatype	77
5.3	Deleting committed datatypes	77
5.4	List of Operations	78
6	Attributes	91
6.1	Creating Attributes	91
6.2	Reading and Writing Data	91
6.3	Listing attributes	91
6.4	Deleting Attributes	91
6.5	List of Operations	92

7	Types	109
7.1	Predefined Types	109
7.2	String Types - Fixed Length	110
7.3	String Types - Variable Length	111
7.4	Compound Types	111
7.5	Enumerated Types	112
7.6	Array Types	113
7.7	Opaque Types	113
7.8	Object Reference Types	114
7.9	Region Reference Types	114
7.10	Type Keys	115
7.11	Related Resources	117
8	Access Control List	119
8.1	Example	119
8.2	Root ACL Inheritance	120
8.3	List of Operations	121
9	Reference	131
9.1	Authorization and Authentication	131
9.2	Common Request Headers	132
9.3	Common Response Headers	132
9.4	Common Error Responses	132
9.5	Diagram of REST operations	133
9.6	Hypermedia	133
9.7	Resource List	134
9.8	Using Iteration	135
10	Indices and tables	137

This is the developer documentation for the HDF REST API.

Contents:

CHAPTER 1

Introduction

The HDF REST API is an interface to HDF5 data stored on network-based architectures. The HDF REST API has provisions to support CRUD (create, read, update and delete) operations on the full spectrum of HDF5 objects including: groups, links, datasets, attributes, and committed data types. See https://support.hdfgroup.org/pubs/papers/RESTful_HDF5.pdf for more information on the RESTful HDF5 interface.

As a REST-based API, implementations can be developed in a variety of languages, such as JavaScript, Python, C, and other common languages. Some services which implement the HDF REST API are:

HDF Kita: <https://www.hdfgroup.org/hdf-kita>

H5serv: <https://github.com/HDFGroup/h5serv>

HDF5 REST VOL: <https://bitbucket.hdfgroup.org/users/jhenderson/repos/rest-vol/browse>

In the HDF REST API, domains are containers for a related collection of resources, similar to a file in the traditional HDF5 library. In the HDF Kita implementation of the HDF REST API, domains *are* files, but in general the HDF REST API supports alternative implementations (e.g. data that is stored in a database). Most operations of the service act on a domain resource that is provided in the *X-Hdf-domain* http header or (alternatively) the domain query parameter.

2.1 Mapping of file paths to domain names

To convert a file path to a domain name:

1. Remove the extension
2. Determine the path relative to the data directory
3. Replace '/' with '.'
4. Reverse the path
5. Add the domain suffix (using the domain config value)

As an example, consider a server installation where the data directory is '/data' and an HDF5 "file" is located at /data/myfolder/an_hdf_file.h5 and hdfgroup.org is the base domain. The above sequence of steps would look like the following:

1. /data/myfolder/an_hdf_file
2. myfolder/an_hdf_file
3. myfolder.an_hdf_file
4. an_hdf_file.myfolder
5. an_hdf_file.myfolder.hdfgroup.org

The final expression is what should be used in the Host field for any request that accesses that file.

For path names that include non-alphanumeric characters, replace any such characters with the string '%XX' where XX is the hexadecimal value of the character. For example:

this.file.has.dots.h5

becomes:

this%2Efile%2Ehase%2Edots

2.2 Creating Domains

Use *PUT Domain* to create a domain. The domain name must follow DNS conventions (e.g. two consecutive “dots” are not allowed). After creation, the domain will contain just one resource, the root group.

2.3 Getting Information about Domains

Use *GET Domain* to retrieve information about a specific domain (specified in the Host header), including the UUID of the domain’s root group. If the Host value is not supplied, the service returns information in the auto-generated Table of Contents (TOC) that provides information on domains that are available.

2.4 Deleting Domains

Use *DELETE Domain* to delete a domain. All resources within the domain will be deleted!

The TOC domain cannot be deleted.

2.5 List of Operations

2.5.1 DELETE Domain

Description

The DELETE operation deletes the given domain and all its resources (groups, datasets, attributes, etc.).

Requests

Syntax

```
DELETE / HTTP/1.1
X-Hdf-domain: DOMAIN
Authorization: <authorization_string>
```

```
DELETE /?domain=DOMAIN HTTP/1.1
Authorization: <authorization_string>
```

Request Parameters

This implementation of the operation does not use request parameters.

Request Headers

This implementation of the operation uses only the request headers that are common to most requests. See *Common Request Headers*.

Responses

Response Headers

This implementation of the operation uses only response headers that are common to most responses. See *Common Response Headers*.

Response Elements

This implementation of the operation does not return any response elements.

Special Errors

This implementation of the operation does not return special errors. For general information on standard error codes, see *Common Error Responses*.

Examples

Sample Request

```
DELETE / HTTP/1.1
Host: hsdshdflab.hdfgroup.org
X-Hdf-domain: /shared/deleteme.h5
Content-Length: 0
Accept: */*
```

Sample cURL command

```
$ curl -X DELETE -u username:password --header "X-Hdf-domain: /shared/deleteme.h5" \
↳ hsdshdflab.hdfgroup.org/
```

Sample Response

```
HTTP/1.1 200 OK
Date: Thu, 12 Jul 2018 16:33:54 GMT
Content-Length: 0
Content-Type: application/json
Server: nginx/1.15.0
```

Related Resources

- *GET Domain*
- *PUT Domain*

2.5.2 GET Domain

Description

This operation retrieves information about the requested domain.

Note: If the HDF Dynamic DNS Server (see <https://github.com/HDFGroup/dynamic-dns>) is running, the operations can specify the domain as part of the URI. Example: <http://tall.data.hdfgroup.org:7253/> returns data about the domain “tall” hosted on data.hdfgroup.org. The DNS server will determine the proper IP that maps to this domain.

If the DNS Server is not setup, specify the desired domain in the “X-Hdf-domain” line of the http header.

Alternatively, the domain can be specified as a query parameter. Example:

<http://127.0.0.1:7253?domain=tall.data.hdfgroup.org>.

If no X-Hdf-domain value is supplied, the default Table of Contents (TOC) domain is returned.

Requests

Syntax

```
GET / HTTP/1.1
X-Hdf-domain: DOMAIN
Authorization: <authorization_string>
```

```
GET /?domain=DOMAIN HTTP/1.1
Authorization: <authorization_string>
```

Request Parameters

This implementation of the operation does not use request parameters.

Request Headers

This implementation of the operation uses only the request headers that are common to most requests. See *Common Request Headers*

Responses

Response Headers

This implementation of the operation uses only response headers that are common to most responses. See *Common Response Headers*.

Response Elements

On success, a JSON response will be returned with the following elements:

root

The UUID of the root group of this domain.

created

A timestamp giving the time the domain was created in UTC (ISO-8601 format).

owner

The user which owns the domain.

lastModified

A timestamp giving the most recent time that any content in the domain has been modified in UTC (ISO-8601 format).

hrefs

An array of links to related resources. See *Hypermedia*.

Special Errors

This implementation of the operation does not return any special errors. For general information on standard error codes, see *Common Error Responses*.

Examples

Sample Request

```
GET / HTTP/1.1
Host: hsdshdflab.hdfgroup.org
X-Hdf-domain: /shared/tall.h5
Accept-Encoding: gzip, deflate
Accept: */*
```

Sample cURL command

```
$ curl -X GET --header "X-Hdf-domain: /shared/tall.h5" hsdshdflab.hdfgroup.org/
```

Sample Response

```
HTTP/1.1 200 OK
Date: Thu, 12 Jul 2018 16:08:23 GMT
Content-Length: 638
Etag: "e45bef255ffc0530c33857b88b15f551f371de38"
Content-Type: application/json
Server: nginx/1.15.0
```

```
{
  "root": "g-be5996fa-83c5-11e8-a8e6-0242ac120016",
  "created": 1531174596.0696847,
  "owner": "admin",
  "class": "domain",
  "lastModified": 1531174596.0696847,
  "hrefs": [
    {"href": "hdsdshdflab.hdfgroup.org/", "rel": "self"},
    {"href": "hdsdshdflab.hdfgroup.org/datasets", "rel": "database"},
    {"href": "hdsdshdflab.hdfgroup.org/groups", "rel": "groupbase"},
    {"href": "hdsdshdflab.hdfgroup.org/datatypes", "rel": "typebase"},
    {"href": "hdsdshdflab.hdfgroup.org/groups/g-be5996fa-83c5-11e8-a8e6-0242ac120016", "rel": "root"},
    {"href": "hdsdshdflab.hdfgroup.org/acls", "rel": "acls"},
    {"href": "hdsdshdflab.hdfgroup.org/?domain=/shared", "rel": "parent"}
  ]
}
```

Related Resources

- *DELETE Domain*
- *GET Group*
- *PUT Domain*

2.5.3 PUT Domain

Description

This operation creates a new domain.

Note: Initially the only object contained in the domain is the root group. Use other PUT and POST operations to create new objects in the domain.

Note: The operation will fail if the domain already exists (a 409 code will be returned).

Requests

Syntax

```
PUT / HTTP/1.1
X-Hdf-domain: DOMAIN
Authorization: <authorization_string>
```

```
PUT /?domain=DOMAIN HTTP/1.1
Authorization: <authorization_string>
```

Request Parameters

This implementation of the operation does not use request parameters.

Request Headers

This implementation of the operation uses only the request headers that are common to most requests. See *Common Request Headers*.

Responses

Response Headers

This implementation of the operation uses only response headers that are common to most responses. See *Common Response Headers*.

Response Elements

On success, a JSON response will be returned with the following elements:

root

The UUID of the root group of this domain.

created

A timestamp giving the time the domain was created in UTC (ISO-8601 format).

owner

The user which owns the domain.

lastModified

A timestamp giving the most recent time that any content in the domain has been modified in UTC (ISO-8601 format).

acls

A JSON object representing the *Access Control List* values for the domain.

Special Errors

This implementation of the operation does not return any special errors. For general information on standard error codes, see *Common Error Responses*.

An http status code of 409 (Conflict) will be returned if the domain already exists.

Examples

Sample Request

```
PUT / HTTP/1.1
Host: hsdshdflab.hdfgroup.org
X-Hdf-domain: /shared/newfile.h5
Content-Length: 0
Accept: */*
Accept-Encoding: gzip, deflate
```

Sample cURL command

```
$ curl -X PUT -u username:password --header "X-Hdf-domain: /shared/newfile.h5" \
↳ hsdshdflab.hdfgroup.org/
```

Sample Response

```
HTTP/1.1 201 Created
Date: Thu, 12 Jul 2018 16:16:34 GMT
Content-Length: 380
Content-Type: application/json
Server: nginx/1.15.0
```

```
{
  "root": "g-a6915d1a-85ef-11e8-8659-0242ac12000c",
  "created": 1531412497.967022,
  "owner": "test_user1",
  "lastModified": 1531412497.967022,
  "acls": {
    "default": {"updateACL": false, "read": true, "delete": false, "update": ↵
↳ false, "create": false, "readACL": false},
    "test_user1": {"updateACL": true, "read": true, "delete": true, "update": ↵
↳ true, "create": true, "readACL": true}
  }
}
```

Related Resources

- *DELETE Domain*
- *GET Group*
- *GET Domain*

Groups are objects that can be used to organize objects within a domain. Groups contain *links* which can reference other objects (datasets, groups or committed datatypes). There are four different types of links that can be used:

- hard: A direct link to a group, dataset, or committed datatype object in the domain.
- soft: A symbolic link that gives a path to an object within the domain (object may or may not be present).
- external: A symbolic link to an object that is external to the domain.
- user-defined: A user-defined link (this implementation only provides title and class for user-defined links).

Groups all have attributes which can be used to store meta-data about the group.

3.1 Creating Groups

Use the *POST Group* operation to create new Groups. Initially the new group will have no links and no attributes.

3.2 Getting information about Groups

Use *GET Group* to get information about a group such as the attribute count, link count and creation and modification times.

To retrieve the UUIDs of all the groups in a domain, use *GET Groups*.

To retrieve the links of a group use *GET Links*. Use *GET Link* to get information about a specific link.

To get a group's attributes, use *GET Attributes*.

3.3 Updating Links

To create a hard, soft, or external link, use *PUT Link*.

To delete a link use *DELETE Link*.

Note: deleting a link doesn't delete the object that it refers to.

3.4 Deleting Groups

Use *DELETE Group* to remove a group. All attributes and links of the group will be deleted.

Note: deleting a group will not delete any objects (datasets or other groups) that the the group's links point to. These objects may become *anonymous*, i.e. they are not referenced by any link, but can still be accessed via a GET request with the object UUID.

3.5 List of Operations

3.5.1 DELETE Group

Description

The implementation of the DELETE operation deletes the group with the UUID given in the URI. All attributes and links of the group will also be deleted. In addition, any links from other groups **TO** the deleted group will be removed.

Note: Groups, datatypes, and datasets that are referenced by the group's links will **not** be deleted. Use the DELETE operation for those objects to remove them.

Requests

Syntax

```
DELETE /groups/<id> HTTP/1.1
X-Hdf-domain: DOMAIN
Authorization: <authorization_string>
```

```
DELETE /groups/<id>?domain=DOMAIN HTTP/1.1
Authorization: <authorization_string>
```

<id> is the UUID of the group to be deleted.

Request Parameters

This implementation of the operation does not use request parameters.

Request Headers

This implementation of the operation uses only the request headers that are common to most requests. See *Common Request Headers*

Responses

Response Headers

This implementation of the operation uses only response headers that are common to most responses. See *Common Response Headers*.

Response Elements

This implementation of the operation does not return any response elements.

Special Errors

This implementation of the operation does not return special errors. For general information on standard error codes, see *Common Error Responses*.

Examples

Sample Request

```
DELETE /groups/g-76de767c-8606-11e8-9cc2-0242ac120008 HTTP/1.1
Host: hsdshdflab.hdfgroup.org
X-Hdf-domain: /shared/tall.h5
Authorization: authorization_string
```

Sample cURL command

```
$ curl -X DELETE -u username:password --header "X-Hdf-domain: /shared/tall.h5" \
↳ hsdshdflab.hdfgroup.org/groups/g-76de767c-8606-11e8-9cc2-0242ac120008
```

Sample Response

```
HTTP/1.1 200 OK
Date: Thu, 12 Jul 2018 19:05:43 GMT
Content-Type: application/json
Server: nginx/1.15.0
```

Related Resources

- *POST Group*
- *GET Group*

3.5.2 DELETE Link

Description

The implementation of the DELETE operation deletes the link named in the URI.

Groups, datatypes, and datasets that are referenced by the link will **not** be deleted. To delete groups, datatypes or datasets, use the appropriate DELETE operation for those objects.

Requests

Syntax

```
DELETE /groups/<id>/links/<name> HTTP/1.1
X-Hdf-domain: DOMAIN
Authorization: <authorization_string>
```

```
DELETE /groups/<id>/links/<name>?domain=DOMAIN HTTP/1.1
Authorization: <authorization_string>
```

- *<id>* is the UUID of the group the link is a member of.
- *<name>* is the URL-encoded name of the link.

Request Parameters

This implementation of the operation does not use request parameters.

Request Headers

This implementation of the operation uses only the request headers that are common to most requests. See *Common Request Headers*.

Responses

Response Headers

This implementation of the operation uses only response headers that are common to most responses. See *Common Response Headers*.

Response Elements

On success, a JSON response will be returned with the following elements:

hrefs

An array of links to related resources. See *Hypermedia*.

Special Errors

An attempt to delete the root group will return 403 - Forbidden. For general information on standard error codes, see *Common Error Responses*.

Examples

Sample Request

```
DELETE /groups/g-b116b6f0-85e9-11e8-9cc2-0242ac120008/links/deleteme HTTP/1.1
Host: hsdshdf1ab.hdfgroup.org
X-Hdf-domain: /shared/tall.h5
Content-Length: 0
Accept: */*
Accept-Encoding: gzip, deflate
```

Sample cURL command

```
$ curl -X DELETE -u username:password --header "X-Hdf-domain: /shared/tall.h5" \
↳ hsdshdf1ab.hdfgroup.org/groups/g-b116b6f0-85e9-11e8-9cc2-0242ac120008/links/deleteme
```

Sample Response

```
HTTP/1.1 200 OK
Date: Thu, 12 Jul 2018 19:45:54 GMT
Content-Length: 12
Content-Type: application/json
Server: nginx/1.15.0
```

```
{"href": []}
```

Related Resources

- *DELETE Dataset*
- *DELETE Datatype*
- *DELETE Group*
- *GET Link*
- *GET Groups*
- *POST Group*

3.5.3 GET Group

Description

Returns information about the group with the UUID given in the URI.

Requests

Syntax

```
GET /groups/<id> HTTP/1.1
X-Hdf-domain: DOMAIN
Authorization: <authorization_string>
```

```
GET /groups/<id>?domain=DOMAIN HTTP/1.1
Authorization: <authorization_string>
```

<id> is the UUID of the requested group.

Request Parameters

include_links

If this request parameter is provided, the links of the group are included in the response.

Request Headers

This implementation of the operation uses only the request headers that are common to most requests. See *Common Request Headers*.

Responses

Response Headers

This implementation of the operation uses only response headers that are common to most responses. See *Common Response Headers*.

Response Elements

On success, a JSON response will be returned with the following elements:

id

The UUID of the requested group.

root

The root group of the domain that the group is within.

attributeCount

The number of attributes belonging to the group.

linkCount

The number of links belonging to the group.

created

A timestamp giving the time the group was created in UTC (ISO-8601 format).

lastModified

A timestamp giving the most recent time the group has been modified (i.e. attributes or links updated) in UTC (ISO-8601 format).

hrefs

An array of hypertext links to related resources. See *Hypermedia*.

Special Errors

This implementation of the operation does not return special errors. For general information on standard error codes, see *Common Error Responses*.

Examples

Sample Request

```
GET /groups/g-be6eb652-83c5-11e8-b9ee-0242ac12000a HTTP/1.1
Host: hsdshdflab.hdfgroup.org
X-Hdf-domain: /shared/tall.h5
Accept-Encoding: gzip, deflate
Accept: */*
```

Sample cURL command

```
$ curl -X GET --header "X-Hdf-domain: /shared/tall.h5" hsdshdflab.hdfgroup.org/groups/
↪g-be6eb652-83c5-11e8-b9ee-0242ac12000a
```

Sample Response

```
HTTP/1.1 200 OK
Date: Thu, 12 Jul 2018 18:27:24 GMT
Content-Length: 711
Etag: "2c410d1c469786f25ed0075571a8e7a3f313cec1"
Content-Type: application/json
Server: nginx/1.15.0
```

```
{
  "id": "g-be6eb652-83c5-11e8-b9ee-0242ac12000a",
  "root": "g-be5996fa-83c5-11e8-a8e6-0242ac120016",
  "linkCount": 2,
  "attributeCount": 0,
  "lastModified": 1531174596.5785563,
  "created": 1531174596.2463753,
  "domain": "/shared/tall.h5",
  "hrefs": [
    {"href": "hdsdshdfiab.hdfgroup.org/groups/g-be6eb652-83c5-11e8-b9ee-
↪0242ac12000a", "rel": "self"},
    {"href": "hdsdshdfiab.hdfgroup.org/groups/g-be6eb652-83c5-11e8-b9ee-
↪0242ac12000a/links", "rel": "links"},
    {"href": "hdsdshdfiab.hdfgroup.org/groups/g-be5996fa-83c5-11e8-a8e6-
↪0242ac120016", "rel": "root"},
    {"href": "hdsdshdfiab.hdfgroup.org/", "rel": "home"},
    {"href": "hdsdshdfiab.hdfgroup.org/groups/g-be6eb652-83c5-11e8-b9ee-
↪0242ac12000a/attributes", "rel": "attributes"}
  ]
}
```

Related Resources

- *DELETE Group*
- *GET Links*
- *GET Groups*
- *POST Group*
- *GET Attribute*

3.5.4 GET Groups

Description

Returns UUIDs for all the groups in a domain (other than the root group).

Requests

Syntax

```
GET /groups HTTP/1.1
X-Hdf-domain: DOMAIN
Authorization: <authorization_string>
```

```
GET /groups?domain=DOMAIN HTTP/1.1
Authorization: <authorization_string>
```

Request Parameters

This implementation of the operation uses the following request parameters (both optional):

Limit

If provided, a positive integer value specifying the maximum number of UUIDs to return.

Marker

If provided, a string value indicating that only UUIDs that occur after the marker value will be returned.

Request Headers

This implementation of the operation uses only the request headers that are common to most requests. See *Common Request Headers*.

Responses

Response Headers

This implementation of the operation uses only response headers that are common to most responses. See *Common Response Headers*.

Response Elements

On success, a JSON response will be returned with the following elements:

groups

An array of UUIDs - one for each group (excluding the root group) in the domain. If the “Marker” and/or “Limit” request parameters are used, a subset of the UUIDs may be returned.

hrefs

An array of hypertext links to related resources. See *Hypermedia*.

Special Errors

This implementation of the operation does not return special errors. For general information on standard error codes, see *Common Error Responses*.

Examples

Sample Request

```
GET /groups HTTP/1.1
Host: hsdshdfllab.hdfgroup.org
X-Hdf-domain: /shared/tall.h5
Accept-Encoding: gzip, deflate
Accept: */*
```

Sample cURL command

```
$ curl -X GET --header "X-Hdf-domain: /shared/tall.h5" hsdshdfllab.hdfgroup.org/groups
```

Sample Response

```
HTTP/1.1 200 OK
Date: Thu, 12 Jul 2018 18:40:30 GMT
Content-Length: 443
Etag: "83575a7865761b6d4eaf5d285ab1de062c49250b"
Content-Type: application/json
Server: nginx/1.15.0
```

```
{
  "groups": [
    "g-be6eb652-83c5-11e8-b9ee-0242ac12000a",
    "g-be836c0a-83c5-11e8-947e-0242ac120014",
    "g-beaaa824-83c5-11e8-a8e6-0242ac120016",
    "g-beb56bba-83c5-11e8-87e1-0242ac12000c",
    "g-bf15f8b8-83c5-11e8-8ad9-0242ac120009"
  ],
  "hrefs": [
    {"href": "hsdshdfllab.hdfgroup.org/groups", "rel": "self"},
    {"href": "hsdshdfllab.hdfgroup.org/groups/g-be5996fa-83c5-11e8-a8e6-
↪0242ac120016", "rel": "root"},
    {"href": "hsdshdfllab.hdfgroup.org/", "rel": "home"}
  ]
}
```

Sample Request with Marker and Limit

This example uses the “Marker” request parameter to return only UUIDs after the given Marker value. The “Limit” request parameter is used to limit the number of UUIDs in the response to 2.

```
GET /groups?Marker=g-be836c0a-83c5-11e8-947e-0242ac120014&Limit=2 HTTP/1.1
Host: hsdshdfllab.hdfgroup.org
X-Hdf-domain: /shared/tall.h5
Accept-Encoding: gzip, deflate
Accept: */*
```

Sample cURL command

URL enclosed in quotes to prevent shell from seeing ampersand

```
$ curl -X GET --header "X-Hdf-domain: /shared/tall.h5" "hdsdshdflab.hdfgroup.org/  
↪groups?Marker=g-be836c0a-83c5-11e8-947e-0242ac120014&Limit=2"
```

Sample Response with Marker and Limit

```
HTTP/1.1 200 OK  
Date: Thu, 12 Jul 2018 18:52:35 GMT  
Content-Length: 317  
Etag: "49221af3436fdaca7e26c74b491ccf8698555f08"  
Content-Type: application/json  
Server: nginx/1.15.0
```

```
{  
  "groups": [  
    "g-beaaa824-83c5-11e8-a8e6-0242ac120016",  
    "g-beb56bba-83c5-11e8-87e1-0242ac12000c"  
  ],  
  "hrefs": [  
    {"href": "hdsdshdflab.hdfgroup.org/groups", "rel": "self"},  
    {"href": "hdsdshdflab.hdfgroup.org/groups/g-be5996fa-83c5-11e8-a8e6-  
↪0242ac120016", "rel": "root"},  
    {"href": "hdsdshdflab.hdfgroup.org/", "rel": "home"}  
  ]  
}
```

Related Resources

- *DELETE Group*
- *GET Links*
- *GET Group*
- *POST Group*

3.5.5 GET Link

Description

Returns information about a Link.

Requests

Syntax

```
GET /groups/<id>/links/<name> HTTP/1.1
X-Hdf-domain: DOMAIN
Authorization: <authorization_string>
```

```
GET /groups/<id>/links/<name>?domain=DOMAIN HTTP/1.1
Authorization: <authorization_string>
```

- *<id>* is the UUID of the group the link is a member of.
- *<name>* is the URL-encoded name of the link.

Request Parameters

This implementation of the operation does not use request parameters.

Request Headers

This implementation of the operation uses only the request headers that are common to most requests. See *Common Request Headers*

Responses

Response Headers

This implementation of the operation uses only response headers that are common to most responses. See *Common Response Headers*.

Response Elements

On success, a JSON response will be returned with the following elements:

link["title"]

The name of the link.

link["collection"]

For hard links, the domain collection for which the object the link points to is a member of. The value will be one of: "groups", "datasets", "datatypes". For symbolic links, this element is not present.

link["class"]

Indicates the type of link. One of the following values will be returned:

- H5L_TYPE_HARD: A direct link to a group, dataset, or committed datatype object in the domain
- H5L_TYPE_SOFT: A symbolic link that gives a path to an object within the domain (object may or may not be present).

- `H5L_TYPE_EXTERNAL`: A symbolic link to an object that is external to the domain
- `H5L_TYPE_UDLINK`: A user-defined link (this implementation only provides title and class for user-defined links)

link["h5path"]

For symbolic links ("`H5L_TYPE_SOFT`" or "`H5L_TYPE_EXTERNAL`"), the path to the resource the link references.

link["h5domain"]

For external links, the path of the external domain containing the object that is linked. *Note:* The domain may or may not exist. Use `GET /` with the domain to verify.

link["id"]

For hard links, the uuid of the object the link points to. For symbolic links this element is not present

created

A timestamp giving the time the link was created in UTC (ISO-8601 format).

lastModified

A timestamp giving the most recent time the group has been modified in UTC (ISO-8601 format).

hrefs

An array of hypertext links to related resources. See *Hypermedia*.

Special Errors

This implementation of the operation does not return special errors. For general information on standard error codes, see *Common Error Responses*.

Examples

Sample Request - Hard Link

```
GET /groups/g-be5996fa-83c5-11e8-a8e6-0242ac120016/links/g1 HTTP/1.1
Host: hsdshdfllab.hdfgroup.org
X-Hdf-domain: /shared/tall.h5
Accept-Encoding: gzip, deflate
Accept: */*
```

Sample cURL command

```
$ curl -X GET --header "X-Hdf-domain: /shared/tall.h5" hsdshdflab.hdfgroup.org/groups/  
↪g-be5996fa-83c5-11e8-a8e6-0242ac120016/links/g1
```

Sample Response - Hard Link

```
HTTP/1.1 200 OK  
Date: Thu, 12 Jul 2018 19:53:43 GMT  
Content-Length: 560  
Etag: "70c5c4f2f7cac9f7f155fe026f4c492f65e3fb8e"  
Content-Type: application/json  
Server: nginx/1.15.0
```

```
{  
  "lastModified": 1531174596.2666101,  
  "created": 1531174596.2666101,  
  "link": {  
    "id": "g-be6eb652-83c5-11e8-b9ee-0242ac12000a",  
    "collection": "groups",  
    "class": "H5L_TYPE_HARD",  
    "title": "g1"  
  },  
  "hrefs": [  
    {"href": "hsdshdflab.hdfgroup.org/groups/g-be5996fa-83c5-11e8-a8e6-  
↪0242ac120016/links/g1", "rel": "self"},  
    {"href": "hsdshdflab.hdfgroup.org/", "rel": "home"},  
    {"href": "hsdshdflab.hdfgroup.org/groups/g-be5996fa-83c5-11e8-a8e6-  
↪0242ac120016", "rel": "owner"},  
    {"href": "hsdshdflab.hdfgroup.org/groups/g-be6eb652-83c5-11e8-b9ee-  
↪0242ac12000a", "rel": "target"}  
  ]  
}
```

Sample Request - Soft Link

```
GET /groups/g-beb56bba-83c5-11e8-87e1-0242ac12000c/links/slink HTTP/1.1  
Host: hsdshdflab.hdfgroup.org  
X-Hdf-domain: /shared/tall.h5  
Accept-Encoding: gzip, deflate  
Accept: */*
```

Sample cURL command

```
$ curl -X GET --header "X-Hdf-domain: /shared/tall.h5" hsdshdflab.hdfgroup.org/groups/  
↪g-beb56bba-83c5-11e8-87e1-0242ac12000c/links/slink
```

Sample Response - Soft Link

```
HTTP/1.1 200 OK
Date: Thu, 12 Jul 2018 20:05:11 GMT
Content-Length: 417
Etag: "7bd777729ac5af261c85c7e3b87ef0045739bf77"
Content-Type: application/json
Server: nginx/1.15.0
```

```
{
  "created": 1531174597.2404227,
  "lastModified": 1531174597.2404227,
  "link": {
    "title": "slink",
    "class": "H5L_TYPE_SOFT",
    "h5path": "somevalue"
  },
  "hrefs": [
    {"rel": "self", "href": "hdsdshdflab.hdfgroup.org/groups/g-beb56bba-83c5-11e8-87e1-0242ac12000c/links/slink"},
    {"rel": "home", "href": "hdsdshdflab.hdfgroup.org/"},
    {"rel": "owner", "href": "hdsdshdflab.hdfgroup.org/groups/g-beb56bba-83c5-11e8-87e1-0242ac12000c"}
  ]
}
```

Sample Request - External Link

```
GET /groups/g-beaaa824-83c5-11e8-a8e6-0242ac120016/links/extlink HTTP/1.1
Host: hdsdshdflab.hdfgroup.org
X-Hdf-domain: /shared/tall.h5
Accept-Encoding: gzip, deflate
Accept: */*
```

Sample cURL command

```
$ curl -X GET --header "X-Hdf-domain: /shared/tall.h5" hdsdshdflab.hdfgroup.org/groups/g-beaaa824-83c5-11e8-a8e6-0242ac120016/links/extlink
```

Sample Response - External Link

```
HTTP/1.1 200 OK
Date: Thu, 12 Jul 2018 20:25:26 GMT
Content-Length: 448
Etag: "1b7a228acdb19f7259ed8a1b3ba4bc442b405ef9"
Content-Type: application/json
Server: nginx/1.15.0
```

```
{
  "hrefs": [
```

(continues on next page)

(continued from previous page)

```

    {"href": "hdsdshdflab.hdfgroup.org/groups/g-beaaa824-83c5-11e8-a8e6-
→0242ac120016/links/extlink", "rel": "self"},
    {"href": "hdsdshdflab.hdfgroup.org/", "rel": "home"},
    {"href": "hdsdshdflab.hdfgroup.org/groups/g-beaaa824-83c5-11e8-a8e6-
→0242ac120016", "rel": "owner"}
  ],
  "lastModified": 1531174596.5889843,
  "created": 1531174596.5889843,
  "link": {
    "h5path": "somepath",
    "title": "extlink",
    "class": "H5L_TYPE_EXTERNAL",
    "h5domain": "somefile"
  }
}

```

Sample Request - User Defined Link

TODO

```

GET /groups/0262c3a6-a069-11e4-8905-3c15c2da029e/links/udlink HTTP/1.1
Host: hdsdshdflab.hdfgroup.org
X-Hdf-domain: /shared/tall.h5
Accept-Encoding: gzip, deflate
Accept: */*

```

Sample cURL command

```
$
```

Sample Response - User Defined Link

```

HTTP/1.1 200 OK
Date: Tue, 20 Jan 2015 05:56:00 GMT
Content-Length: 576
Etag: "2ab310eba3bb4282f84d643fcc30e591da485576"
Content-Type: application/json
Server: nginx/1.15.0

```

```

{
  "link": {
    "class": "H5L_TYPE_USER_DEFINED",
    "title": "udlink"
  },
  "created": "2015-01-16T03:47:22Z",
  "lastModified": "2015-01-16T03:47:22Z",
  "hrefs": [
    {"href": "http://tall_with_udlink.test.hdfgroup.org/groups/0262c3a6-a069-11e4-
→8905-3c15c2da029e/links/udlink", "rel": "self"},
    {"href": "http://tall_with_udlink.test.hdfgroup.org/groups/0260b214-a069-11e4-
→a840-3c15c2da029e", "rel": "root"},

```

(continues on next page)

(continued from previous page)

```
{
  "href": "http://tall_with_udlink.test.hdfgroup.org/", "rel": "home"},
  "href": "http://tall_with_udlink.test.hdfgroup.org/groups/0262c3a6-a069-11e4-
↪8905-3c15c2da029e", "rel": "owner"}
]
```

- *DELETE Link*
- *GET Links*
- *PUT Link*

3.5.6 GET Links

Description

Returns all the links for a given group.

Requests

Syntax

```
GET /groups/<id>/links HTTP/1.1
X-Hdf-domain: DOMAIN
Authorization: <authorization_string>
```

```
GET /groups/<id>/links?domain=DOMAIN HTTP/1.1
Authorization: <authorization_string>
```

- *<id>* is the UUID of the group the links to be returned are a member of.

Request Parameters

This implementation of the operation uses the following request parameters (both optional):

Limit

If provided, a positive integer value specifying the maximum number of links to return.

Marker

If provided, a string value indicating that only links that occur after the marker value will be returned.

Request Headers

This implementation of the operation uses only the request headers that are common to most requests. See *Common Request Headers*

Responses

Response Headers

This implementation of the operation uses only response headers that are common to most responses. See *Common Response Headers*.

Response Elements

On success, a JSON response will be returned with the following elements:

links

An array of JSON objects giving information about each link returned. See *GET Link* for a description of the link response elements.

hrefs

An array of links to related resources. See *Hypermedia*.

Special Errors

This implementation of the operation does not return special errors. For general information on standard error codes, see *Common Error Responses*.

Examples

Sample Request

```
GET /groups/g-be5996fa-83c5-11e8-a8e6-0242ac120016/links HTTP/1.1
Host: hsdshdflab.hdfgroup.org
X-Hdf-domain: /shared/tall.h5
Accept-Encoding: gzip, deflate
Accept: */*
```

Sample cURL command

```
$ curl -X GET --header "X-Hdf-domain: /shared/tall.h5" hsdshdflab.hdfgroup.org/groups/
↪g-be5996fa-83c5-11e8-a8e6-0242ac120016/links
```

Sample Response

```

HTTP/1.1 200 OK
Date: Thu, 12 Jul 2018 20:34:59 GMT
Content-Length: 916
Etag: "49edccea8f724108d41d52c98002d6255286ff8"
Content-Type: application/json
Server: nginx/1.15.0

```

```

{
  "links": [
    {
      "id": "g-be6eb652-83c5-11e8-b9ee-0242ac12000a",
      "title": "g1",
      "class": "H5L_TYPE_HARD",
      "collection": "groups",
      "created": 1531174596.2666101,
      "href": "hdsdshdflab.hdfgroup.org/groups/g-be5996fa-83c5-11e8-a8e6-
↪0242ac120016/links/g1",
      "target": "hdsdshdflab.hdfgroup.org/groups/g-be6eb652-83c5-11e8-b9ee-
↪0242ac12000a"
    },
    {
      "id": "g-bf15f8b8-83c5-11e8-8ad9-0242ac120009",
      "title": "g2",
      "class": "H5L_TYPE_HARD",
      "collection": "groups",
      "created": 1531174597.2827835,
      "href": "hdsdshdflab.hdfgroup.org/groups/g-be5996fa-83c5-11e8-a8e6-
↪0242ac120016/links/g2",
      "target": "hdsdshdflab.hdfgroup.org/groups/g-bf15f8b8-83c5-11e8-8ad9-
↪0242ac120009"
    }
  ],
  "hrefs": [
    {"rel": "self", "href": "hdsdshdflab.hdfgroup.org/groups/g-be5996fa-83c5-11e8-
↪a8e6-0242ac120016/links"},
    {"rel": "home", "href": "hdsdshdflab.hdfgroup.org/"},
    {"rel": "owner", "href": "hdsdshdflab.hdfgroup.org/groups/g-be5996fa-83c5-11e8-
↪a8e6-0242ac120016"}
  ]
}

```

Sample Request Batch

```

GET /groups/g-be5996fa-83c5-11e8-a8e6-0242ac120016/links?Marker=g1&Limit=1 HTTP/1.1
Host: hdsdshdflab.hdfgroup.org
X-Hdf-domain: /shared/tall.h5
Accept-Encoding: gzip, deflate
Accept: */*

```

Sample cURL command

URL enclosed in quotes to prevent shell from seeing ampersand

```
$ curl -X GET --header "X-Hdf-domain: /shared/tall.h5" "hdsdshdflab.hdfgroup.org/groups/g-be5996fa-83c5-11e8-a8e6-0242ac120016/links?Marker=g1&Limit=1"
```

Sample Response Batch

```
HTTP/1.1 200 OK
Date: Thu, 12 Jul 2018 20:41:25 GMT
Content-Length: 597
Etag: "221affdeae54076d3493ce8ce0ed80ddb89c6e27"
Content-Type: application/json
Server: nginx/1.15.0
```

```
{
  "links": [
    {
      "id": "g-bf15f8b8-83c5-11e8-8ad9-0242ac120009",
      "title": "g2",
      "class": "H5L_TYPE_HARD",
      "collection": "groups",
      "created": 1531174597.2827835,
      "href": "hdsdshdflab.hdfgroup.org/groups/g-be5996fa-83c5-11e8-a8e6-0242ac120016/links/g2",
      "target": "hdsdshdflab.hdfgroup.org/groups/g-bf15f8b8-83c5-11e8-8ad9-0242ac120009"
    }
  ],
  "hrefs": [
    {"rel": "self", "href": "hdsdshdflab.hdfgroup.org/groups/g-be5996fa-83c5-11e8-a8e6-0242ac120016/links"},
    {"rel": "home", "href": "hdsdshdflab.hdfgroup.org/"},
    {"rel": "owner", "href": "hdsdshdflab.hdfgroup.org/groups/g-be5996fa-83c5-11e8-a8e6-0242ac120016"}
  ]
}
```

Related Resources

- [DELETE Link](#)
- [GET Link](#)
- [GET Group](#)
- [PUT Link](#)

3.5.7 POST Group

Description

Creates a new Group.

Note: By default the new Group will not be linked from any other group in the domain. A link element can be included in the request body to have an existing group link to the new group. Alternatively, use the *PUT link* operation to link the new group.

Requests

Syntax

```
POST /groups HTTP/1.1
X-Hdf-domain: DOMAIN
Authorization: <authorization_string>
```

```
POST /groups?domain=DOMAIN HTTP/1.1
Authorization: <authorization_string>
```

Request Parameters

This implementation of the operation does not use request parameters.

Request Headers

This implementation of the operation uses only the request headers that are common to most requests. See *Common Request Headers*

Request Elements

Optionally the request body can be a JSON object that has a link key with sub-keys:

id

The UUID of the group that the new group should be linked to. If the UUID is not valid, the request will fail and a new group will not be created.

name

The name of the new link.

Responses

Response Headers

This implementation of the operation uses only response headers that are common to most responses. See *Common Response Headers*.

Response Elements

On success, a JSON response will be returned with the following elements:

id

The UUID of the newly created group.

root

The root group of the domain which the group was created in.

attributeCount

The number of attributes belonging to the group.

linkCount

The number of links belonging to the group.

created

A timestamp giving the time the group was created in UTC (ISO-8601 format).

lastModified

A timestamp giving the most recent time the group has been modified (i.e. attributes or links updated) in UTC (ISO-8601 format).

Special Errors

This implementation of the operation does not return special errors. For general information on standard error codes, see *Common Error Responses*.

Examples

Sample Request

Create a new, un-linked Group.

```
POST /groups HTTP/1.1
Host: hsdshdflab.hdfgroup.org
X-Hdf-domain: /shared/tall.h5
Content-Length: 0
Accept: */*
Accept-Encoding: gzip, deflate
```

Sample cURL command

```
$ curl -X POST -u username:password --header "X-Hdf-domain: /shared/tall.h5" \
↳ hsdshdflab.hdfgroup.org/groups
```

Sample Response

```
HTTP/1.1 201 Created
Content-Length: 202
Server: nginx/1.15.0
Date: Thu, 12 Jul 2018 16:49:10 GMT
Content-Type: application/json
```

```
{
  "id": "g-7fbbf52a-85f3-11e8-9cc2-0242ac120008",
  "root": "g-b116b6f0-85e9-11e8-9cc2-0242ac120008",
  "created": 1531414150.1522243,
  "lastModified": 1531414150.1522243,
  "linkCount": 0,
  "attributeCount": 0
}
```

Sample Request with Link

Create a new Group, link to root (which has uuid of “g-b116b6f0-...”) as “linked_group”.

```
POST /groups HTTP/1.1
Host: hsdshdflab.hdfgroup.org
X-Hdf-domain: /shared/tall.h5
Content-Length: 82
Accept: */*
Accept-Encoding: gzip, deflate
```

```
{
  "link": {
    "id": "g-b116b6f0-85e9-11e8-9cc2-0242ac120008",
    "name": "linked_group"
  }
}
```

Sample cURL command

```
$ curl -X POST -u username:password --header "X-Hdf-domain: /shared/tall.h5" \
-d "{\"link\": {\"id\": \"g-b116b6f0-85e9-11e8-9cc2-0242ac120008\", \"name\": \
↳ \"linked_group\"}}\" hsdshdflab.hdfgroup.org/groups
```

Sample Response with Link

```
HTTP/1.1 201 Created
Content-Length: 200
Server: nginx/1.15.0
Date: Thu, 12 Jul 2018 16:57:57 GMT
Content-Type: application/json
```

```
{
  "id": "g-b9bd362a-85f4-11e8-a549-0242ac12000b",
  "root": "g-b116b6f0-85e9-11e8-9cc2-0242ac120008",
  "linkCount": 0,
  "attributeCount": 0,
  "lastModified": 1531414676.963812,
  "created": 1531414676.963812
}
```

Related Resources

- *DELETE Group*
- *GET Links*
- *PUT Link*
- *GET Group*
- *GET Groups*

3.5.8 PUT Link

Description

Creates a new link in a given group.

Either hard, soft, or external links can be created based on the request elements. See examples below.

Note: any existing link with the same name will be replaced with the new link.

Requests

Syntax

```
PUT /groups/<id>/links/<name> HTTP/1.1
X-Hdf-domain: DOMAIN
Authorization: <authorization_string>
```

```
PUT /groups/<id>/links/<name>?domain=DOMAIN HTTP/1.1
Authorization: <authorization_string>
```

- *<id>* is the UUID of the group that the link will be created in.
- *<name>* is the URL-encoded name of the link.

Request Parameters

This implementation of the operation does not use request parameters.

Request Headers

This implementation of the operation uses only the request headers that are common to most requests. See *Common Request Headers*

Request Elements

The request body must include a JSON object that has the following keys:

id

For Hard links - The UUID of the object to link to. If the UUID is not valid, the request will fail and a new link will not be created. If this key is present, the h5path and h5domain keys will be ignored.

h5path

For Soft and External links - A string describing a path to an external resource. If this key is present a soft or external link will be created.

h5domain

For External links - A string giving the external domain where the resource is present. If this key is present, the h5path key must be provided as well.

Responses

Response Headers

This implementation of the operation uses only response headers that are common to most responses. See *Common Response Headers*.

Response Elements

On success, a JSON response will be returned with the following elements:

hrefs

An array of links to related resources. See *Hypermedia*.

Special Errors

This implementation of the operation does not return special errors. For general information on standard error codes, see *Common Error Responses*.

Examples

Sample Request - Create Hard Link

In group “g-b116b6f0-...”, create a hard link named “g3” that points to the object with UUID “g-b9bd362a-...”.

```
PUT /groups/g-b116b6f0-85e9-11e8-9cc2-0242ac120008/links/g3 HTTP/1.1
Host: hsdshdf1ab.hdfgroup.org
X-Hdf-domain: /shared/tall.h5
Content-Length: 48
Content-Type: application/json
Accept: */*
Accept-Encoding: gzip, deflate
```

```
{"id": "g-b9bd362a-85f4-11e8-a549-0242ac12000b"}
```

Sample cURL command

```
$ curl -X PUT -u username:password --header "X-Hdf-domain: /shared/tall.h5" --header
↪ "Content-Type: application/json"
  -d "{\"id\": \"g-b9bd362a-85f4-11e8-a549-0242ac12000b\"}" hsdshdf1ab.hdfgroup.org/
↪ groups/g-b116b6f0-85e9-11e8-9cc2-0242ac120008/links/g3
```

Sample Response - Create Hard Link

```
HTTP/1.1 201 Created
Date: Thu, 12 Jul 2018 19:25:16 GMT
Content-Length: 13
Content-Type: application/json
Server: nginx/1.15.0
```

```
{"hrefs": []}
```

Sample Request - Create Soft Link

In group “g-b116b6f0-...”, create a soft link named “softlink” that contains the path “/somewhere”.

```
PUT /groups/g-b116b6f0-85e9-11e8-9cc2-0242ac120008/links/softlink HTTP/1.1
Host: hsdshdf1ab.hdfgroup.org
X-Hdf-domain: /shared/tall.h5
Content-Length: 24
Accept: */*
Accept-Encoding: gzip, deflate
```

```
{"h5path": "/somewhere"}
```

Sample cURL command

```
$ curl -X PUT -u username:password --header "X-Hdf-domain: /shared/tall.h5" --header
↳ "Content-Type: application/json"
  -d "{\"h5path\": \"/somewhere\"}" hsdshdfllab.hdfgroup.org/groups/g-b116b6f0-85e9-
↳ 11e8-9cc2-0242ac120008/links/softlink
```

Sample Response - Create Soft Link

```
HTTP/1.1 201 Created
Date: Thu, 12 Jul 2018 19:30:28 GMT
Content-Length: 13
Content-Type: application/json
Server: nginx/1.15.0
```

```
{"hrefs": []}
```

Sample Request - Create External Link

In group “g-b116b6f0-...”, create an external link named “extlink” that references the object at path: “/dset1” in domain: “/shared/ext_file.h5”.

```
PUT /groups/g-b116b6f0-85e9-11e8-9cc2-0242ac120008/links/extlink HTTP/1.1
Host: hsdshdfllab.hdfgroup.org
X-Hdf-domain: /shared/tall.h5
Content-Length: 51
Accept: */*
Accept-Encoding: gzip, deflate
```

```
{"h5domain": "/shared/ext_file.h5", "h5path": "/dset1"}
```

Sample cURL command

```
$ curl -X PUT -u username:password --header "X-Hdf-domain: /shared/tall.h5" --header
↳ "Content-Type: application/json"
  -d "{\"h5domain\": \"/shared/ext_file.h5\", \"h5path\": \"/dset1\"}" hsdshdfllab.
↳ hdfgroup.org/groups/g-b116b6f0-85e9-11e8-9cc2-0242ac120008/links/extlink
```

Sample Response - Create External Link

```
HTTP/1.1 201 Created
Date: Thu, 12 Jul 2018 19:34:57 GMT
Content-Length: 13
Content-Type: application/json
Server: nginx/1.15.0
```

```
{"hrefs": []}
```

Related Resources

- *DELETE Link*
- *GET Link*
- *GET Links*
- *GET Group*

Datasets are objects that are composed of a homogenous collection of data elements. Each dataset has a *type* that specifies the structure of the individual elements (float, string, compound, etc.), and a *shape* that specifies the layout of the data elements (scalar, one-dimensional, multi-dimensional). In addition, meta-data can be attached to a dataset in the form of attributes. See: *Attributes*.

4.1 Creating Datasets

Use the *POST Dataset* operation to create new datasets. As part of the POST request, JSON descriptions for the type and shape of the dataset are included with the request. Optionally, creation properties can be used to specify the chunk layout (how the data elements are stored in the server) and compression filters (e.g. GZIP, LZF, SZIP).

4.2 Getting information about a dataset

Use the *GET Dataset* operation to retrieve information about a dataset's type, shape, creation properties, and number of attributes. To list all the datasets within a domain use *GET Datasets*. To list the datasets linked to a particular group use *GET Links* and look at links with a "collection" key of "datasets".

4.3 Writing data to a dataset

To write data into the dataset, use the *PUT Value* operation. The request can either provide values for the entire dataset, or values for a hyperslab (rectangular sub-region) selection. In addition, if it desired to update a specific list of data elements, a point selection (series of element coordinates) can be passed to the *PUT Value* operation.

4.4 Reading data from a dataset

To read either the entire dataset, or a specified selection, use the *GET Value* operation. Without any request parameters, the GET operation returns all data values. To read a specific hyperslab, use the `select` parameter to specify the start and end indices of the hyperslab (the selection can also include a `step` value to include a regular subset of the hyperslab). Finally, for one-dimensional datasets with compound types, a *where* parameter can be used to select elements meeting a specified condition.

To read a specific list of elements (by index values), use the *POST Value* operation (POST is used in this case rather than GET since the point selection values may be too large to include in the URI.)

4.5 Resizable datasets

If one or more of the dimensions of a dataset may need to be extended after creation, provide a *maxdims* key to the shape during creation (see *POST Dataset*). If the value of the *maxdims* dimension is 0, that dimension is *unlimited* and may be extended as much as desired. If an upper limit is known, use that value in *maxdims* to allow that dimension to be extended up to the given value. To resize the dataset, use the *PUT Shape* operation with the desired shape value(s) for the new dimensions.

Note: dimensions can only be increased, not decreased.

4.6 Deleting datasets

The *DELETE Dataset* operation will remove the dataset, its attributes, and any links to the object.

4.7 List of Operations

4.7.1 DELETE Dataset

Description

The implementation of the DELETE operation deletes the dataset named in the URI. All attributes and links of the dataset will also be deleted. In addition any links from other groups to the deleted group will be removed.

Requests

Syntax

```
DELETE /datasets/<id> HTTP/1.1
Host: DOMAIN
Authorization: <authorization_string>
```

<id> is the UUID of the requested dataset to be deleted.

Request Parameters

This implementation of the operation does not use request parameters.

Request Headers

This implementation of the operation uses only the request headers that are common to most requests. See *Common Request Headers*.

Responses

Response Headers

This implementation of the operation uses only response headers that are common to most responses. See *Common Response Headers*.

Response Elements

On success, a JSON response will be returned with the following elements:

hrefs

An array of links to related resources. See *Hypermedia*.

Special Errors

This implementation of the operation does not return special errors. For general information on standard error codes, see *Common Error Responses*.

Examples

Sample Request

```
DELETE /datasets/289bb654-a2c6-11e4-97d8-3c15c2da029e HTTP/1.1
Content-Length: 0
User-Agent: python-requests/2.3.0 CPython/2.7.8 Darwin/14.0.0
host: tall_dset112_deleted.test.hdfgroup.org
Accept: */*
Accept-Encoding: gzip, deflate
```

Sample Response

```
HTTP/1.1 200 OK
Date: Fri, 23 Jan 2015 06:07:49 GMT
Content-Length: 287
Content-Type: application/json
Server: TornadoServer/3.2.2
```

```
{
  "hrefs": [
    {"href": "http://tall_dset112_deleted.test.hdfgroup.org/datasets", "rel": "self"},
    {"href": "http://tall_dset112_deleted.test.hdfgroup.org/groups/289b4873-a2c6-11e4-
↪adfb-3c15c2da029e", "rel": "root"},
    {"href": "http://tall_dset112_deleted.test.hdfgroup.org/", "rel": "home"}
  ]
}
```

Related Resources

- *GET Datasets*
- *GET Dataset*
- *POST Dataset*

4.7.2 GET Dataset

Description

Returns information about the dataset with the UUID given in the URI.

Requests

Syntax

```
GET /datasets/<id> HTTP/1.1
Host: DOMAIN
Authorization: <authorization_string>
```

<id> is the UUID of the requested dataset.

Request Parameters

This implementation of the operation does not use request parameters.

Request Headers

This implementation of the operation uses only the request headers that are common to most requests. See *Common Request Headers*

Responses

Response Headers

This implementation of the operation uses only response headers that are common to most responses. See *Common Response Headers*.

Response Elements

On success, a JSON response will be returned with the following elements:

id

The UUID of the dataset object.

type

A JSON object representing the type of the dataset. See *Types* for details of the type representation.

shape

A JSON object representing the shape of the dataset. See *GET Shape* for details of the shape representation.

creationProperties

A JSON object that describes chunk layout, filters, fill value, and other aspects of the dataset. See: <http://hdf5-json.readthedocs.org/en/latest/bnf/dataset.html#grammar-token-dcpl> for a complete description of fields that can be used.

attributeCount

The number of attributes belonging to the dataset.

created

A timestamp giving the time the dataset was created in UTC (ISO-8601 format).

lastModified

A timestamp giving the most recent time the group has been modified (i.e. attributes or links updated) in UTC (ISO-8601 format).

hrefs

An array of links to related resources. See *Hypermedia*.

Special Errors

This implementation of the operation does not return special errors. For general information on standard error codes, see *Common Error Responses*.

Examples

Sample Request

```
GET /datasets/c8d83759-a2c6-11e4-8713-3c15c2da029e HTTP/1.1
host: tall.test.hdfgroup.org
Accept-Encoding: gzip, deflate
Accept: */*
User-Agent: python-requests/2.3.0 CPython/2.7.8 Darwin/14.0.0
```

Sample Response

```
HTTP/1.1 200 OK
Date: Fri, 23 Jan 2015 06:15:33 GMT
Content-Length: 755
Etag: "ecbd7e52654b0a8f4ccbebac06175ce5df5f8c79"
Content-Type: application/json
Server: TornadoServer/3.2.2
```

```
{
  "id": "c8d83759-a2c6-11e4-8713-3c15c2da029e",
  "shape": {
    "dims": [10],
    "class": "H5S_SIMPLE"
  },
  "type": {
    "base": "H5T_IEEE_F32BE",
    "class": "H5T_FLOAT"
  },
  "creationProperties": {
    "allocTime": "H5D_ALLOC_TIME_LATE",
    "fillTime": "H5D_FILL_TIME_IFSET",
    "layout": {
      "class": "H5D_CONTIGUOUS"
    }
  },
  "attributeCount": 0,
  "created": "2015-01-23T06:12:18Z",
  "lastModified": "2015-01-23T06:12:18Z",
  "hrefs": [
    {"href": "http://tall.test.hdfgroup.org/datasets/c8d83759-a2c6-11e4-8713-↵
↵3c15c2da029e", "rel": "self"},
    {"href": "http://tall.test.hdfgroup.org/groups/c8d7842b-a2c6-11e4-b4f1-↵
↵3c15c2da029e", "rel": "root"},
    {"href": "http://tall.test.hdfgroup.org/datasets/c8d83759-a2c6-11e4-8713-↵
↵3c15c2da029e/attributes", "rel": "attributes"},
    {"href": "http://tall.test.hdfgroup.org/datasets/c8d83759-a2c6-11e4-8713-↵
↵3c15c2da029e/value", "rel": "data"},
    {"href": "http://tall.test.hdfgroup.org/", "rel": "home"}
  ]
}
```

Related Resources

- *DELETE Dataset*
- *GET Attributes*
- *GET Shape*
- *GET Type*
- *GET Datasets*
- *GET Value*
- *POST Value*
- *PUT Value*

4.7.3 GET Datasets

Description

Returns UUIDs for all the datasets in a domain.

Requests

Syntax

```
GET /datasets HTTP/1.1
Host: DOMAIN
Authorization: <authorization_string>
```

Request Parameters

This implementation of the operation uses the following request parameters (both optional):

Limit

If provided, a positive integer value specifying the maximum number of UUIDs to return.

Marker

If provided, a string value indicating that only UUIDs that occur after the marker value will be returned.

Request Headers

This implementation of the operation uses only the request headers that are common to most requests. See *Common Request Headers*

Responses

Response Headers

This implementation of the operation uses only response headers that are common to most responses. See *Common Response Headers*.

Response Elements

On success, a JSON response will be returned with the following elements:

datasets

An array of UUIDs, one for each dataset in the domain.

hrefs

An array of links to related resources. See *Hypermedia*.

Special Errors

This implementation of the operation does not return special errors. For general information on standard error codes, see *Common Error Responses*.

Examples

Sample Request

```
GET /datasets HTTP/1.1
host: tall.test.hdfgroup.org
Accept-Encoding: gzip, deflate
Accept: */*
User-Agent: python-requests/2.3.0 CPython/2.7.8 Darwin/14.0.0
```

Sample Response

```
HTTP/1.1 200 OK
Date: Fri, 23 Jan 2015 06:33:36 GMT
Content-Length: 413
Etag: "977e96c7bc63a6e05d10d56565df2ab8d30e404d"
Content-Type: application/json
Server: TornadoServer/3.2.2
```

```
{
  "datasets": [
    "c8d7dd14-a2c6-11e4-a68c-3c15c2da029e",
    "c8d7f159-a2c6-11e4-99af-3c15c2da029e",
```

(continues on next page)

(continued from previous page)

```

    "c8d83759-a2c6-11e4-8713-3c15c2da029e",
    "c8d84a8a-a2c6-11e4-b457-3c15c2da029e"
  ],
  "hrefs": [
    {"href": "http://tall.test.hdfgroup.org/datasets", "rel": "self"},
    {"href": "http://tall.test.hdfgroup.org/groups/c8d7842b-a2c6-11e4-b4f1-
↪3c15c2da029e", "rel": "root"},
    {"href": "http://tall.test.hdfgroup.org/", "rel": "home"}
  ]
}

```

Sample Request with Marker and Limit

This example uses the “Marker” request parameter to return only UUIDs after the given Marker value. The “Limit” request parameter is used to limit the number of UUIDs in the response to 5.

```

GET /datasets?Marker=817db263-a2cc-11e4-87f2-3c15c2da029e&Limit=5 HTTP/1.1
host: dset1k.test.hdfgroup.org
Accept-Encoding: gzip, deflate
Accept: */*
User-Agent: python-requests/2.3.0 CPython/2.7.8 Darwin/14.0.0

```

Sample Response with Marker and Limit

```

HTTP/1.1 200 OK
Date: Fri, 23 Jan 2015 06:53:52 GMT
Content-Length: 459
Etag: "cb708d4839cc1e165fe6bb30718e49589ef140f4"
Content-Type: application/json
Server: TornadoServer/3.2.2

```

```

{
  "datasets": [
    "817dcfb8-a2cc-11e4-9197-3c15c2da029e",
    "817de9ee-a2cc-11e4-8378-3c15c2da029e",
    "817e028a-a2cc-11e4-8ce3-3c15c2da029e",
    "817e1b61-a2cc-11e4-ba39-3c15c2da029e",
    "817e341c-a2cc-11e4-a16f-3c15c2da029e"
  ],
  "hrefs": [
    {"href": "http://dset1k.test.hdfgroup.org/datasets", "rel": "self"},
    {"href": "http://dset1k.test.hdfgroup.org/groups/81760a80-a2cc-11e4-bb55-
↪3c15c2da029e", "rel": "root"},
    {"href": "http://dset1k.test.hdfgroup.org/", "rel": "home"}
  ]
}

```

Related Resources

- *DELETE Dataset*
- *GET Dataset*

- *POST Dataset*

4.7.4 GET Shape

Description

Gets the shape of a dataset.

Requests

Syntax

```
GET /datasets/<id>/shape HTTP/1.1
Host: DOMAIN
Authorization: <authorization_string>
```

<id> is the UUID of the dataset that shape is requested for.

Request Parameters

This implementation of the operation does not use request parameters.

Request Headers

This implementation of the operation uses only the request headers that are common to most requests. See *Common Request Headers*.

Responses

Response Headers

This implementation of the operation uses only response headers that are common to most responses. See *Common Response Headers*.

Response Elements

On success, a JSON response will be returned with the following elements:

shape

A JSON object with the following keys:

class: A string with one of the following values:

- H5S_NULL: A null dataspace, which has no elements
- H5S_SCALAR: A dataspace with a single element (although possibly of a complex datatype)
- H5S_SIMPLE: A dataspace that consists of a regular array of elements

dims: An integer array whose length is equal to the number of dimensions (rank) of the dataspace. The value of each element gives the current size of each dimension. Dims is not returned for H5S_NULL or H5S_SCALAR dataspaces.

maxdims: An integer array whose length is equal to the number of dimensions of the dataspace. The value of each element gives the maximum size of each dimension. A value of 0 indicates that the dimension has *unlimited* extent. maxdims is not returned for H5S_SIMPLE dataspaces which are not extensible or for H5S_NULL or H5S_SCALAR dataspaces.

fillvalue: A value, of a type compatible with the dataset's type, which gives the *fill* value for the dataset (the value which elements will be initialized to when a dataspace is extended). fillvalue is only returned for extensible dataspaces.

created

A timestamp giving the time the datashape (same as the dataset) was created in UTC (ISO-8601 format).

lastModified

A timestamp giving the most recent time the dataspace has been modified (i.e. a dimension has been extended) in UTC (ISO-8601 format).

hrefs

An array of links to related resources. See *Hypermedia*.

Special Errors

This implementation of the operation does not return special errors. For general information on standard error codes, see *Common Error Responses*.

Examples

Sample Request

```
GET /datasets/3b57b6d4-a6a8-11e4-96b5-3c15c2da029e/shape HTTP/1.1
host: tall.test.hdfgroup.org
Accept-Encoding: gzip, deflate
Accept: */*
User-Agent: python-requests/2.3.0 CPython/2.7.8 Darwin/14.0.0
```

Sample Response

```
HTTP/1.1 200 OK
Date: Wed, 28 Jan 2015 04:43:41 GMT
Content-Length: 445
Etag: "76ed777f151c70d0560d1414bffe1515a3df86b0"
Content-Type: application/json
Server: TornadoServer/3.2.2
```

```
{
  "shape": {
    "class": "H5S_SIMPLE"
    "dims": [10],
  },
  "created": "2015-01-28T04:40:23Z",
  "lastModified": "2015-01-28T04:40:23Z",
  "hrefs": [
    {"href": "http://tall.test.hdfgroup.org/datasets/3b57b6d4-a6a8-11e4-96b5-
↪3c15c2da029e", "rel": "self"},
    {"href": "http://tall.test.hdfgroup.org/datasets/3b57b6d4-a6a8-11e4-96b5-
↪3c15c2da029e", "rel": "owner"},
    {"href": "http://tall.test.hdfgroup.org/groups/3b56ee54-a6a8-11e4-b2ae-
↪3c15c2da029e", "rel": "root"}
  ],
}
```

Sample Request - Resizable

```
GET /datasets/a64010e8-a6aa-11e4-98c8-3c15c2da029e/shape HTTP/1.1
host: resizable.test.hdfgroup.org
Accept-Encoding: gzip, deflate
Accept: */*
User-Agent: python-requests/2.3.0 CPython/2.7.8 Darwin/14.0.0
```

Sample Response - Resizable

```
HTTP/1.1 200 OK
Date: Wed, 28 Jan 2015 05:00:59 GMT
Content-Length: 500
Etag: "1082800980d6809a8008b22e225f1adde8afc73f"
Content-Type: application/json
Server: TornadoServer/3.2.2
```

```
{
  "shape": {
    "class": "H5S_SIMPLE",
    "dims": [10, 10],
    "maxdims": [10, 0],
  },
  "created": "2015-01-28T04:40:23Z",
  "lastModified": "2015-01-28T04:40:23Z",
  "hrefs": [
    {"href": "http://resizable.test.hdfgroup.org/datasets/a64010e8-a6aa-11e4-98c8-
↪3c15c2da029e", "rel": "self"},
    {"href": "http://resizable.test.hdfgroup.org/datasets/a64010e8-a6aa-11e4-98c8-
↪3c15c2da029e", "rel": "owner"},
    {"href": "http://resizable.test.hdfgroup.org/groups/a63f5dcf-a6aa-11e4-ab68-
↪3c15c2da029e", "rel": "root"}
  ]
}
```

Related Resources

- [GET Dataset](#)
- [GET Type](#)
- [PUT Shape](#)

4.7.5 GET Type

Description

Gets Type Information for a dataset.

Requests

Syntax

```
GET /datasets/<id>/type HTTP/1.1
Host: DOMAIN
Authorization: <authorization_string>
```

<id> is the UUID of the dataset the type information is requested for.

Request Parameters

This implementation of the operation does not use request parameters.

Request Headers

This implementation of the operation uses only the request headers that are common to most requests. See [Common Request Headers](#)

Responses

Response Headers

This implementation of the operation uses only response headers that are common to most responses. See [Common Response Headers](#).

Response Elements

On success, a JSON response will be returned with the following elements:

type

A JSON object representing the type definition for the dataset. See [Types](#) for information on how different types are represented.

hrefs

An array of links to related resources. See *Hypermedia*.

Special Errors

This implementation of the operation does not return special errors. For general information on standard error codes, see *Common Error Responses*.

Examples

Sample Request - Predefined Type

```
GET /datasets/ba06ce68-a6b5-11e4-8ed3-3c15c2da029e/type HTTP/1.1
host: scalar.test.hdfgroup.org
Accept-Encoding: gzip, deflate
Accept: */*
User-Agent: python-requests/2.3.0 CPython/2.7.8 Darwin/14.0.0
```

Sample Response - Predefined Type

```
HTTP/1.1 200 OK
Date: Wed, 28 Jan 2015 06:20:16 GMT
Content-Length: 519
Etag: "802b160bf786596a9cb9f6d5cd6faa4fe1127e8c"
Content-Type: application/json
Server: TornadoServer/3.2.2
```

```
{
  "type": {
    "class": "H5T_INTEGER",
    "order": "H5T_ORDER_LE",
    "base_size": 4,
    "base": "H5T_STD_I32LE",
    "size": 4
  },
  "hrefs": [
    { "href": "http://scalar.test.hdfgroup.org/datasets/ba06ce68-a6b5-11e4-8ed3-
    ↪3c15c2da029e/type", "rel": "self" },
    { "href": "http://scalar.test.hdfgroup.org/datasets/ba06ce68-a6b5-11e4-8ed3-
    ↪3c15c2da029e", "rel": "owner" },
    { "href": "http://scalar.test.hdfgroup.org/groups/ba06992e-a6b5-11e4-9ba5-
    ↪3c15c2da029e", "rel": "root" }
  ]
}
```

Sample Request - Compound Type

```
GET /datasets/b9edddd7-a6b5-11e4-9afd-3c15c2da029e/type HTTP/1.1
host: compound.test.hdfgroup.org
Accept-Encoding: gzip, deflate
Accept: */*
User-Agent: python-requests/2.3.0 CPython/2.7.8 Darwin/14.0.0
```

Sample Response - Compound Type

```
HTTP/1.1 200 OK
Date: Wed, 28 Jan 2015 06:20:16 GMT
Content-Length: 1199
Etag: "1f97eac24aa18d3c462a2f2797c4782a1f2a0aa2"
Content-Type: application/json
Server: TornadoServer/3.2.2
```

```
{
  "type": {
    "class": "H5T_COMPOUND",
    "fields": [
      {
        "type": {
          "order": "H5T_ORDER_LE",
          "base_size": 8,
          "class": "H5T_INTEGER",
          "base": "H5T_STD_I64LE",
          "size": 8},
        "name": "date"
      }, {
        "type": {
          "strpad": "H5T_STR_NULLPAD",
          "base_size": 6, "order": "H5T_ORDER_NONE",
          "cset": "H5T_CSET_ASCII",
          "strsize": 6,
          "class": "H5T_STRING",
          "size": 6},
        "name": "time"
      }, {
        "type": {
          "order": "H5T_ORDER_LE",
          "base_size": 8,
          "class": "H5T_INTEGER",
          "base": "H5T_STD_I64LE",
          "size": 8},
        "name": "temp"
      }, {
        "type": {
          "order": "H5T_ORDER_LE",
          "base_size": 8,
          "class": "H5T_FLOAT",
          "base": "H5T_IEEE_F64LE",
          "size": 8},
        "name": "pressure"
      }, {
        "type": {
          "strpad": "H5T_STR_NULLPAD",
```

(continues on next page)

(continued from previous page)

```

        "base_size": 6,
        "order": "H5T_ORDER_NONE",
        "cset": "H5T_CSET_ASCII",
        "strsize": 6,
        "class": "H5T_STRING",
        "size": 6,
        "name": "wind"}
    ]
},
"hrefs": [
  {"href": "http://compound.test.hdfgroup.org/datasets/b9edddd7-a6b5-11e4-9afd-
↪3c15c2da029e/type", "rel": "self"},
  {"href": "http://compound.test.hdfgroup.org/datasets/b9edddd7-a6b5-11e4-9afd-
↪3c15c2da029e", "rel": "owner"},
  {"href": "http://compound.test.hdfgroup.org/groups/b9eda805-a6b5-11e4-aa52-
↪3c15c2da029e", "rel": "root"}
]
}

```

Related Resources

- *GET Dataset*
- *GET Shape*
- *POST Dataset*

4.7.6 GET Value

Description

Gets data values of a dataset.

Requests

Syntax

```

GET /datasets/<id>/value HTTP/1.1
Host: DOMAIN
Authorization: <authorization_string>

```

<id> is the UUID of the requested dataset.

Request Parameters

select

Optionally the request can provide a select value to indicate a hyperslab selection for the values to be returned - i.e. a rectangular (in 1, 2, or more dimensions) region of the dataset. Format is the following as a url-encoded value:

```
[dim1_start:dim1_end:dim1_step, dim2_start:dim2_end:dim2_step, ... , dimn_start:dimn_stop:dimn_step]
```

The number of tuples “start:stop:step” should equal the number of dimensions of the dataset.

For each tuple:

- start must be greater than equal to zero and less than the dimension extent
- stop must be greater than or equal to start and less than or equal to the dimension extent
- step is optional and if provided must be greater than 0. If not provided, the step value for that dimension is assumed to be 1.

query

Optionally the request can provide a query value to select items from a dataset based on a condition expression. E.g. The condition: “(temp > 32.0) & (dir == ‘N’)” would return elements of the dataset where the ‘temp’ field was greater than 32.0 and the ‘dir’ field was equal to ‘N’.

Note: the query value needs to be url-encoded.

Note: the query parameter can be used in conjunction with the select parameter to restrict the return set to the provided selection.

Note: the query parameter can be used in conjunction with the Limit parameter to limit the number of matches returned.

Note: Currently the query parameter can only be used with compound type datasets that are one-dimensional.

Limit

If provided, a positive integer value specifying the maximum number of elements to return. Only has an effect if used in conjunction with the query parameter.

Request Headers

This implementation of the operation supports the common headers in addition to the “Accept” header value of “application/octet-stream”. Use this accept value if a binary response is desired. Binary data will be more efficient for large data requests. If a binary response can be returned, the “Content-Type” response header will be “application/octet-stream”. Otherwise the response header will be “json”.

Note: Binary responses are only supported for datasets that have a fixed-length type (i.e. either a fixed length primitive type or compound type that in turn consists of fixed-length types). Namely variable length strings and variable length data types will always be returned as JSON.

Note: if a binary response is returned, it will consist of the equivalent binary data of the “data” item in the JSON response. No data representing “hrefs” is returned.

For other request headers, see *Common Request Headers*

Responses

Response Headers

This implementation of the operation uses only response headers that are common to most responses. See *Common Response Headers*.

Response Elements

On success, a JSON response will be returned with the following elements:

value

A json array (integer or string for scalar datasets) giving the values of the requested dataset region.

index

A list of indices for each element that met the query condition (only provided when the query request parameter is used).

hrefs

An array of links to related resources. See *Hypermedia*.

Special Errors

This implementation of the operation does not return special errors. For general information on standard error codes, see *Common Error Responses*.

Examples

Sample Request

```
GET /datasets/548f2f21-a83c-11e4-8baf-3c15c2da029e/value HTTP/1.1
host: tall.test.hdfgroup.org
Accept-Encoding: gzip, deflate
Accept: */*
User-Agent: python-requests/2.3.0 CPython/2.7.8 Darwin/14.0.0
```

Sample Response

```
HTTP/1.1 200 OK
Date: Fri, 30 Jan 2015 04:56:20 GMT
Content-Length: 776
Etag: "788efb3caaba7fd2ae5d1edb40b474ba94c877a8"
Content-Type: application/json
Server: TornadoServer/3.2.2
```

```
{
  "value": [
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 1, 2, 3, 4, 5, 6, 7, 8, 9],
    [0, 2, 4, 6, 8, 10, 12, 14, 16, 18],
    [0, 3, 6, 9, 12, 15, 18, 21, 24, 27],
```

(continues on next page)

(continued from previous page)

```

    [0, 4, 8, 12, 16, 20, 24, 28, 32, 36],
    [0, 5, 10, 15, 20, 25, 30, 35, 40, 45],
    [0, 6, 12, 18, 24, 30, 36, 42, 48, 54],
    [0, 7, 14, 21, 28, 35, 42, 49, 56, 63],
    [0, 8, 16, 24, 32, 40, 48, 56, 64, 72],
    [0, 9, 18, 27, 36, 45, 54, 63, 72, 81]
  ],
  "hrefs": [
    {"href": "http://tall.test.hdfgroup.org/datasets/548f2f21-a83c-11e4-8baf-
↪3c15c2da029e/value", "rel": "self"},
    {"href": "http://tall.test.hdfgroup.org/groups/548ed535-a83c-11e4-b58b-
↪3c15c2da029e", "rel": "root"},
    {"href": "http://tall.test.hdfgroup.org/datasets/548f2f21-a83c-11e4-8baf-
↪3c15c2da029e", "rel": "owner"},
    {"href": "http://tall.test.hdfgroup.org/", "rel": "home"}
  ]
}

```

Sample Request - Selection

```

GET /datasets/a299db70-ab57-11e4-9c00-3c15c2da029e/value?select=[1:9,1:9:2] HTTP/1.1
host: tall.test.hdfgroup.org
Accept-Encoding: gzip, deflate
Accept: */*
User-Agent: python-requests/2.3.0 CPython/2.7.8 Darwin/14.0.0

```

Sample Response - Selection

```

HTTP/1.1 200 OK
Date: Tue, 03 Feb 2015 04:01:41 GMT
Content-Length: 529
Etag: "b370a3d34bdd7ebf57a496bc7f0da7bc5a1aafb9"
Content-Type: application/json
Server: TornadoServer/3.2.2

```

```

{
  "value": [
    [1, 3, 5, 7],
    [2, 6, 10, 14],
    [3, 9, 15, 21],
    [4, 12, 20, 28],
    [5, 15, 25, 35],
    [6, 18, 30, 42],
    [7, 21, 35, 49],
    [8, 24, 40, 56]
  ],
  "hrefs": [
    {"href": "http://tall.test.hdfgroup.org/datasets/a299db70-ab57-11e4-9c00-
↪3c15c2da029e/value", "rel": "self"},
    {"href": "http://tall.test.hdfgroup.org/groups/a29982cf-ab57-11e4-b976-
↪3c15c2da029e", "rel": "root"},
    {"href": "http://tall.test.hdfgroup.org/datasets/a299db70-ab57-11e4-9c00-
↪3c15c2da029e", "rel": "owner"},

```

(continues on next page)

(continued from previous page)

```
{
  "href": "http://tall.test.hdfgroup.org/", "rel": "home"
}
```

Sample Request - Query

Get elements from dataset where the 'date' field is equal to 20 and the 'temp' field is greater than or equal to 70.

```
GET /datasets/b2c82938-0e2e-11e5-9092-3c15c2da029e/value?query=(date%20==%2021)%20%26
↪%20(temp%20%3E=%2072) HTTP/1.1
host: compound.test.hdfgroup.org
Accept-Encoding: gzip, deflate
Accept: */*
User-Agent: python-requests/2.3.0 CPython/2.7.8 Darwin/14.0.0
```

Sample Response - Query

```
HTTP/1.1 200 OK
Date: Thu, 11 Jun 2015 21:05:06 GMT
Content-Length: 805
Etag: "927b5ed89616896d3dce7df8bddac058321076a"
Content-Type: application/json
Server: TornadoServer/4.1
```

```
{
  "index": [68, 69, 70, 71],
  "value": [
    [21, "17:53", 74, 29.87, "S 9"],
    [21, "16:53", 75, 29.87, "SW 10"],
    [21, "15:53", 79, 29.87, "S 12"],
    [21, "14:53", 78, 29.87, "SW 9"]
  ]
},
"hrefs": [
  {"href": "http://compound.test.hdfgroup.org/datasets/b2c82938-0e2e-11e5-9092-
↪3c15c2da029e/value", "rel": "self"},
  {"href": "http://compound.test.hdfgroup.org/groups/b2c7f935-0e2e-11e5-96ae-
↪3c15c2da029e", "rel": "root"},
  {"href": "http://compound.test.hdfgroup.org/datasets/b2c82938-0e2e-11e5-9092-
↪3c15c2da029e", "rel": "owner"},
  {"href": "http://compound.test.hdfgroup.org/", "rel": "home"}
]
```

Sample Request - Query Batch

Get elements where the 'date' field is equal to 23 and the index is between 24 and 72. Limit the number of results to 5.

```
GET /datasets/b2c82938-0e2e-11e5-9092-3c15c2da029e/value?query=date%20==%2023&Limit=5&
↪select=[24:72] HTTP/1.1
host: compound.test.hdfgroup.org
Accept-Encoding: gzip, deflate
Accept: */*
User-Agent: python-requests/2.3.0 CPython/2.7.8 Darwin/14.0.0
```

Sample Response - Query Batch

```
HTTP/1.1 200 OK
Date: Thu, 11 Jun 2015 21:15:28 GMT
Content-Length: 610
Etag: "927b5ed89616896d3dce7df8bdddac058321076a"
Content-Type: application/json
Server: TornadoServer/4.1
```

```
{
  "index": [24, 25, 26, 27, 28],
  "value": [
    [23, "13:53", 65, 29.83, "W 5"],
    [23, "12:53", 66, 29.84, "W 5"],
    [23, "11:53", 64, 29.84, "E 6"],
    [23, "10:53", 61, 29.86, "SE 5"],
    [23, "9:53", 62, 29.86, "S 6"]
  ],
  "hrefs": [
    {"href": "http://compound.test.hdfgroup.org/datasets/b2c82938-0e2e-11e5-9092-
↪3c15c2da029e/value", "rel": "self"},
    {"href": "http://compound.test.hdfgroup.org/groups/b2c7f935-0e2e-11e5-96ae-
↪3c15c2da029e", "rel": "root"},
    {"href": "http://compound.test.hdfgroup.org/datasets/b2c82938-0e2e-11e5-9092-
↪3c15c2da029e", "rel": "owner"},
    {"href": "http://compound.test.hdfgroup.org/", "rel": "home"}
  ]
}
```

Related Resources

- *GET Dataset*
- *POST Value*
- *PUT Value*

4.7.7 POST Dataset

Description

Creates a new Dataset.

Requests

Syntax

```
POST /datasets HTTP/1.1
Host: DOMAIN
Authorization: <authorization_string>
```

Request Parameters

This implementation of the operation does not use request parameters.

Request Headers

This implementation of the operation uses only the request headers that are common to most requests. See *Common Request Headers*

Request Elements

The request body must include a JSON object with a “type” key. Optionally “shape”, “maxdims”, and “link” keys can be provided.

type

Either a string that is one of the predefined type values, a UUID of a committed type, or a JSON object describing the type. See *Types* for details of the type specification.

shape

Either a string with the value `H5S_NULL` or an integer array describing the initial dimensions of the dataset. If shape is not provided, a scalar dataset will be created. If the shape value of `H5S_NULL` is specified a dataset with a null dataspace will be created. A null dataset has attributes and a type, but will not be able to store any values.

maxdims

An integer array describing the maximum extent of each dimension (or 0 for unlimited dimensions). If maxdims is not provided the resulting dataset will be non-extensible. Not valid to include if `H5S_NULL` is specified for the shape.

creationProperties

A JSON object that can specify chunk layout, filters, fill value, and other aspects of the dataset. See: <http://hdf5-json.readthedocs.org/en/latest/bnf/dataset.html#grammar-token-dcpl> for a complete description of fields that can be used.

If creationProperties is not provided, default values will be used.

link["id"]

The UUID of the group that the new dataset should be linked to. If the UUID is not valid, the request will fail and a new dataset will not be created.

link["name"]

The name of the new link.

Responses**Response Headers**

This implementation of the operation uses only response headers that are common to most responses. See *Common Response Headers*.

Response Elements

On success, a JSON response will be returned with the following elements:

id

The UUID of the newly created dataset.

attributeCount

The number of attributes belonging to the dataset.

created

A timestamp giving the time the dataset was created in UTC (ISO-8601 format).

lastModified

A timestamp giving the most recent time the dataset has been modified (i.e. attributes or links updated) in UTC (ISO-8601 format).

hrefs

An array of links to related resources. See *Hypermedia*.

Special Errors

This implementation of the operation does not return special errors. For general information on standard error codes, see *Common Error Responses*.

Examples

Sample Request

Create a one-dimensional dataset with 10 floating point elements.

```
POST /datasets HTTP/1.1
Content-Length: 39
User-Agent: python-requests/2.3.0 CPython/2.7.8 Darwin/14.0.0
host: newdset.datasettest.test.hdfgroup.org
Accept: */*
Accept-Encoding: gzip, deflate
```

```
{
  "shape": 10,
  "type": "H5T_IEEE_F32LE"
}
```

Sample Response

```
HTTP/1.1 201 Created
Date: Thu, 29 Jan 2015 06:14:02 GMT
Content-Length: 651
Content-Type: application/json
Server: TornadoServer/3.2.2
```

```
{
  "id": "0568d8c5-a77e-11e4-9f7a-3c15c2da029e",
  "attributeCount": 0,
  "created": "2015-01-29T06:14:02Z",
  "lastModified": "2015-01-29T06:14:02Z",
  "hrefs": [
    {"href": "http://newdset.datasettest.test.hdfgroup.org/datasets/0568d8c5-a77e-11e4-9f7a-3c15c2da029e", "rel": "self"},
    {"href": "http://newdset.datasettest.test.hdfgroup.org/groups/055fe7de-a77e-11e4-bbe9-3c15c2da029e", "rel": "root"},
    {"href": "http://newdset.datasettest.test.hdfgroup.org/datasets/0568d8c5-a77e-11e4-9f7a-3c15c2da029e/attributes", "rel": "attributes"},
    {"href": "http://newdset.datasettest.test.hdfgroup.org/datasets/0568d8c5-a77e-11e4-9f7a-3c15c2da029e/value", "rel": "value"}
  ]
}
```

Sample Request with Link

Create a dataset with 10 variable length string elements. Create link in group: “5e441dcf-...” with name: “linked_dset”.

```
POST /datasets HTTP/1.1
Content-Length: 235
User-Agent: python-requests/2.3.0 CPython/2.7.8 Darwin/14.0.0
host: newdsetwithlink.datasettest.test.hdfgroup.org
```

(continues on next page)

(continued from previous page)

```
Accept: */*
Accept-Encoding: gzip, deflate
```

```
{
  "type": {
    "class": "H5T_STRING",
    "strsize": "H5T_VARIABLE",
    "cset": "H5T_CSET_ASCII",
    "order": "H5T_ORDER_NONE",
    "strpad": "H5T_STR_NULLTERM"
  },
  "shape": 10,
  "link": {
    "id": "5e441dcf-a782-11e4-bd6b-3c15c2da029e",
    "name": "linked_dset"
  }
}
```

Sample Response with Link

```
HTTP/1.1 201 Created
Date: Thu, 29 Jan 2015 06:45:09 GMT
Content-Length: 683
Content-Type: application/json
Server: TornadoServer/3.2.2
```

```
{
  "id": "5e579297-a782-11e4-93f9-3c15c2da029e",
  "attributeCount": 0,
  "created": "2015-01-29T06:45:09Z",
  "lastModified": "2015-01-29T06:45:09Z",
  "hrefs": [
    {"href": "http://newdsetwithlink.datasettest.test.hdfgroup.org/datasets/5e579297-a782-11e4-93f9-3c15c2da029e", "rel": "self"},
    {"href": "http://newdsetwithlink.datasettest.test.hdfgroup.org/groups/5e441dcf-a782-11e4-bd6b-3c15c2da029e", "rel": "root"},
    {"href": "http://newdsetwithlink.datasettest.test.hdfgroup.org/datasets/5e579297-a782-11e4-93f9-3c15c2da029e/attributes", "rel": "attributes"},
    {"href": "http://newdsetwithlink.datasettest.test.hdfgroup.org/datasets/5e579297-a782-11e4-93f9-3c15c2da029e/value", "rel": "value"}
  ]
}
```

Sample Request - Resizable Dataset

Create a one-dimensional dataset with 10 elements, but extendable to an unlimited dimension.

```
POST /datasets HTTP/1.1
Content-Length: 54
User-Agent: python-requests/2.3.0 CPython/2.7.8 Darwin/14.0.0
host: resizabledataset.datasettest.test.hdfgroup.org
```

(continues on next page)

(continued from previous page)

```
Accept: */*
Accept-Encoding: gzip, deflate
```

```
{
  "type": "H5T_IEEE_F32LE",
  "shape": 10,
  "maxdims": 0
}
```

Sample Response - Resizable Dataset

```
HTTP/1.1 201 Created
Date: Thu, 29 Jan 2015 08:28:19 GMT
Content-Length: 675
Content-Type: application/json
Server: TornadoServer/3.2.2
```

```
{
  "id": "c79933ab-a790-11e4-b36d-3c15c2da029e",
  "attributeCount": 0,
  "created": "2015-01-29T08:28:19Z",
  "lastModified": "2015-01-29T08:28:19Z",
  "hrefs": [
    { "href": "http://resizabledataset.datasettest.test.hdfgroup.org/datasets/c79933ab-
    ↪a790-11e4-b36d-3c15c2da029e", "rel": "self" },
    { "href": "http://resizabledataset.datasettest.test.hdfgroup.org/groups/c7759c11-
    ↪a790-11e4-ae03-3c15c2da029e", "rel": "root" },
    { "href": "http://resizabledataset.datasettest.test.hdfgroup.org/datasets/c79933ab-
    ↪a790-11e4-b36d-3c15c2da029e/attributes", "rel": "attributes" },
    { "href": "http://resizabledataset.datasettest.test.hdfgroup.org/datasets/c79933ab-
    ↪a790-11e4-b36d-3c15c2da029e/value", "rel": "value" }
  ]
}
```

Sample Request - Committed Type

Create a two-dimensional dataset which uses a committed type with UUID:

```
POST /datasets HTTP/1.1
Content-Length: 67
User-Agent: python-requests/2.3.0 CPython/2.7.8 Darwin/14.0.0
host: committedtype.datasettest.test.hdfgroup.org
Accept: */*
Accept-Encoding: gzip, deflate
```

```
{
  "type": "accd0b1e-a792-11e4-bada-3c15c2da029e",
  "shape": [10, 10]
}
```

Sample Response - Committed Type

```
HTTP/1.1 201 Created
Date: Thu, 29 Jan 2015 08:41:53 GMT
Content-Length: 675
Content-Type: application/json
Server: TornadoServer/3.2.2
```

```
{
  "id": "ace8cdca-a792-11e4-ad88-3c15c2da029e",
  "attributeCount": 0,
  "created": "2015-01-29T08:41:53Z",
  "lastModified": "2015-01-29T08:41:53Z",
  "hrefs": [
    {"href": "http://committedtype.datasettest.test.hdfgroup.org/datasets/ace8cdca-
    ↪a792-11e4-ad88-3c15c2da029e", "rel": "self"},
    {"href": "http://committedtype.datasettest.test.hdfgroup.org/groups/acc4d37d-a792-
    ↪11e4-b326-3c15c2da029e", "rel": "root"},
    {"href": "http://committedtype.datasettest.test.hdfgroup.org/datasets/ace8cdca-
    ↪a792-11e4-ad88-3c15c2da029e/attributes", "rel": "attributes"},
    {"href": "http://committedtype.datasettest.test.hdfgroup.org/datasets/ace8cdca-
    ↪a792-11e4-ad88-3c15c2da029e/value", "rel": "value"}
  ]
}
```

Sample Request - SZIP Compression with chunking

```
POST /datasets HTTP/1.1
Content-Length: 67
User-Agent: python-requests/2.3.0 CPython/2.7.8 Darwin/14.0.0
host: szip.datasettest.test.hdfgroup.org
Accept: */*
Accept-Encoding: gzip, deflate
```

```
{
  "creationProperties": {
    "filters": [
      {
        "bitsPerPixel": 8,
        "coding": "H5_SZIP_EC_OPTION_MASK",
        "id": 4,
        "pixelsPerBlock": 32,
        "pixelsPerScanline": 100
      }
    ],
    "layout": {
      "class": "H5D_CHUNKED",
      "dims": [
        100,
        100
      ]
    }
  },
  "shape": [
```

(continues on next page)

(continued from previous page)

```
    1000,  
    1000  
  ],  
  "type": "H5T_IEEE_F32LE"  
}
```

Sample Response - SZIP Compression with chunking

```
HTTP/1.1 201 Created  
Date: Thu, 18 Jun 2015 08:41:53 GMT  
Content-Length: 975  
Content-Type: application/json  
Server: TornadoServer/3.2.2
```

```
{  
  "id": "ad283c05-158c-11e5-bd67-3c15c2da029e",  
  "attributeCount": 0,  
  "created": "2015-06-18T07:36:04Z",  
  "lastModified": "2015-06-18T07:36:04Z",  
  "hrefs": [  
    {  
      "href": "http://newdset_szip.datasettest.test.hdfgroup.org/datasets/ad283c05-  
↪158c-11e5-bd67-3c15c2da029e",  
      "rel": "self"  
    },  
    {  
      "href": "http://newdset_szip.datasettest.test.hdfgroup.org/groups/ad2746d4-  
↪158c-11e5-a083-3c15c2da029e",  
      "rel": "root"  
    },  
    {  
      "href": "http://newdset_szip.datasettest.test.hdfgroup.org/datasets/ad283c05-  
↪158c-11e5-bd67-3c15c2da029e/attributes",  
      "rel": "attributes"  
    },  
    {  
      "href": "http://newdset_szip.datasettest.test.hdfgroup.org/datasets/ad283c05-  
↪158c-11e5-bd67-3c15c2da029e/value",  
      "rel": "value"  
    }  
  ]  
}
```

Related Resources

- *GET Dataset*
- *GET Datasets*
- *GET Value*
- *POST Value*
- *PUT Value*

4.7.8 POST Value

Description

Gets values of a dataset for a given point selection (provided in the body of the request).

Requests

Syntax

```
POST /datasets/<id>/value HTTP/1.1
Host: DOMAIN
Authorization: <authorization_string>
```

<id> is the UUID of the requested dataset t

Request Parameters

This implementation of the operation does not use request parameters.

Request Headers

This implementation of the operation uses only the request headers that are common to most requests. See *Common Request Headers*

Request Body

The request body should be a JSON object with the following key:

points

An array of points defining the selection. Each point can either be an integer (if the dataset has just one dimension), or an array where the length of the array is equal to the number of dimensions of the dataset.

Responses

Response Headers

This implementation of the operation uses only response headers that are common to most responses. See *Common Response Headers*.

Response Elements

On success, a JSON response will be returned with the following elements:

value

An array of values where the length of the array is equal to the number of points in the request. Each value will be a string, integer, or JSON object consistent with the dataset type (e.g. a compound type).

Special Errors

This implementation of the operation does not return special errors. For general information on standard error codes, see *Common Error Responses*.

Examples

Sample Request

```
POST /datasets/4e83ad1c-ab6e-11e4-babb-3c15c2da029e/value HTTP/1.1
Content-Length: 92
User-Agent: python-requests/2.3.0 CPython/2.7.8 Darwin/14.0.0
host: tall.test.hdfgroup.org
Accept: */*
Accept-Encoding: gzip, deflate
```

```
{
  "points": [19, 17, 13, 11, 7, 5, 3, 2]
}
```

Sample Response

```
HTTP/1.1 200 OK
Date: Tue, 03 Feb 2015 06:31:38 GMT
Content-Length: 47
Content-Type: application/json
Server: TornadoServer/3.2.2
```

```
{
  "value": [0, 1, 4, 9, 16, 25, 36, 49, 64, 81]
}
```

Related Resources

- *GET Dataset*
- *GET Value*
- *PUT Value*

4.7.9 PUT Shape

Description

Modifies the dimensions of a dataset. Dimensions can only be changed if the dataset was initially created with that dimension as *extensible* - i.e. the maxdims value for that dimension is larger than the initial dimension size (or maxdims set to 0).

Note: Dimensions can only be made larger, they can not be reduced.

Requests

Syntax

```
PUT /datasets/<id>/shape HTTP/1.1
Host: DOMAIN
Authorization: <authorization_string>
```

<id> is the UUID of the dataset whose shape will be modified.

Request Parameters

This implementation of the operation does not use request parameters.

Request Headers

This implementation of the operation uses only the request headers that are common to most requests. See *Common Request Headers*

Request Elements

The request body must include a JSON object with a “shape” key as described below:

shape

An integer array giving the new dimensions of the dataset.

Responses

Response Headers

This implementation of the operation uses only response headers that are common to most responses. See *Common Response Headers*.

Response Elements

On success, a JSON response will be returned with the following elements:

hrefs

An array of links to related resources. See *Hypermedia*.

Special Errors

This implementation of the operation does not return special errors. For general information on standard error codes, see *Common Error Responses*.

Examples

Sample Request

```
PUT /datasets/b9b6acc0-a839-11e4-aa86-3c15c2da029e/shape HTTP/1.1
Content-Length: 19
User-Agent: python-requests/2.3.0 CPython/2.7.8 Darwin/14.0.0
host: resized.test.hdfgroup.org
Accept: */*
Accept-Encoding: gzip, deflate
```

```
{
  "shape": [10, 25]
}
```

Sample Response

```
HTTP/1.1 201 Created
Date: Fri, 30 Jan 2015 04:47:47 GMT
Content-Length: 331
Content-Type: application/json
Server: TornadoServer/3.2.2
```

```
{
  "hrefs": [
    { "href": "http://resized.test.hdfgroup.org/datasets/22e1b235-a83b-11e4-97f4-
    ↪3c15c2da029e", "rel": "self" },
    { "href": "http://resized.test.hdfgroup.org/datasets/22e1b235-a83b-11e4-97f4-
    ↪3c15c2da029e", "rel": "owner" },
    { "href": "http://resized.test.hdfgroup.org/groups/22dfff8f-a83b-11e4-883d-
    ↪3c15c2da029e", "rel": "root" }
  ]
}
```

Related Resources

- *GET Dataset*
- *GET Shape*
- *GET Value*

- *POST Value*
- *PUT Value*

4.7.10 PUT Value

Description

Update the values in a dataset.

Requests

Syntax

```
PUT /datasets/<id>/value HTTP/1.1
Host: DOMAIN
Authorization: <authorization_string>
```

<id> is the UUID of the requested dataset.

Request Parameters

This implementation of the operation does not use request parameters.

Request Headers

This implementation of the operation uses only the request headers that are common to most requests. See *Common Request Headers*

Request Body

The request body should be a JSON object with the following keys:

start:

An optional key that gives the starting coordinate of the selection to be updated. The start value can either be an integer (for 1 dimensional arrays) or an array of integers where the length of the array is equal to the number of dimensions of the dataset. Each value must be greater than or equal to zero and less than the extent of the corresponding dimension.

If start is not provided, the selection starts at 0 for each dimension.

stop:

An optional key that gives the ending coordinate of the selection to be updated. The stop value can either be an integer (for 1 dimensional arrays) or an array of integers where the length of the array is equal to the number of dimensions of the dataset. Each value must be greater than or equal to start (or zero if start is not provided) and less than the extent of the corresponding dimension.

step:

An optional key that gives the step value (i.e. the increment of the coordinate for each supplied value). The step value can either be an integer (for 1 dimensional arrays) or an array of integers where the length of the array is equal to the number of dimensions of the dataset. Each value must be greater than or equal to start (or zero if start is not provided) and less than or equal to the extent of the corresponding dimension.

points:

An optional key that contains a list of array elements to be updated. Each element of the list should be an integer if the dataset is of rank 1 or an n-element list (where n is the dataset rank) if the dataset rank is greater than 1. If points is provided (indicating a point selection update), then start, stop, and step (used for hyperslab selection) should not be provided.

value:

A JSON array containing the data values to be written.

value_base64:

Use this key instead of “value” to use base64-encoded binary data rather than JSON ascii. This will be more efficient for large data transfers than using a JSON array.

Note: “value_base64” is only supported for fixed length datatypes.

Responses

Response Headers

This implementation of the operation uses only response headers that are common to most responses. See *Common Response Headers*.

Response Elements

No response elements are returned.

Special Errors

This implementation of the operation does not return special errors. For general information on standard error codes, see *Common Error Responses*.

Examples

Sample Request

This example writes a 10x10 integer dataset with the values 0-99 inclusive.

```
PUT /datasets/817e2280-ab5d-11e4-afe6-3c15c2da029e/value HTTP/1.1
Content-Length: 465
User-Agent: python-requests/2.3.0 CPython/2.7.8 Darwin/14.0.0
host: valueput.datasettest.test.hdfgroup.org
Accept: */*
Accept-Encoding: gzip, deflate
```

```
{
  "value": [
    [0, 1, 2, 3, 4, 5, 6, 7, 8, 9],
    [10, 11, 12, 13, 14, 15, 16, 17, 18, 19],
    [20, 21, 22, 23, 24, 25, 26, 27, 28, 29],
    [30, 31, 32, 33, 34, 35, 36, 37, 38, 39],
    [40, 41, 42, 43, 44, 45, 46, 47, 48, 49],
    [50, 51, 52, 53, 54, 55, 56, 57, 58, 59],
    [60, 61, 62, 63, 64, 65, 66, 67, 68, 69],
    [70, 71, 72, 73, 74, 75, 76, 77, 78, 79],
    [80, 81, 82, 83, 84, 85, 86, 87, 88, 89],
    [90, 91, 92, 93, 94, 95, 96, 97, 98, 99]
  ]
}
```

Sample Response

```
HTTP/1.1 200 OK
Date: Tue, 03 Feb 2015 04:31:22 GMT
Content-Length: 0
Content-Type: text/html; charset=UTF-8
Server: TornadoServer/3.2.2
```

Sample Request - Selection

This example writes a portion of the dataset by using the start and stop keys in the request.

```
PUT /datasets/b2d0af00-ab65-11e4-a874-3c15c2da029e/value HTTP/1.1
Content-Length: 92
User-Agent: python-requests/2.3.0 CPython/2.7.8 Darwin/14.0.0
host: valueputsel.datasettest.test.hdfgroup.org
Accept: */*
Accept-Encoding: gzip, deflate
```

```
{
  "start": 5,
  "stop": 10,
  "value": [13, 17, 19, 23, 29]
}
```

Sample Response

```
HTTP/1.1 200 OK
Date: Tue, 03 Feb 2015 05:30:01 GMT
Content-Length: 0
Content-Type: text/html; charset=UTF-8
Server: TornadoServer/3.2.2
```

Related Resources

- *GET Dataset*
- *GET Value*
- *POST Value*

Committed Datatypes

Committed datatypes (also known as “named types”), are objects that describe types. These types can be used in the creation of datasets and attributes.

Committed datatypes can be linked to from a Group and can contain attributes, just like a dataset or group object.

5.1 Creating committed datatypes

Use *POST Datatype* to create a new datatype. A complete description of the type must be sent with the POST request.

5.2 Getting information about a committed datatype

Use the *GET Datatype* operation to retrieve information about a committed datatype. To list all the committed datatypes within a domain use *GET Datatypes*. To list the committed types linked to a particular group use *GET Links* and examine link objects with a “collection” key of “datatypes”.

5.3 Deleting committed datatypes

Use *DELETE Datatype* to delete a datatype. Links from any group to the datatype will be deleted.

5.4 List of Operations

5.4.1 DELETE Datatype

Description

The implementation of the **DELETE** operation deletes the committed datatype named in the URI. All attributes of the datatype will also be deleted.

Requests

Syntax

```
DELETE /datatypes/<id> HTTP/1.1
X-Hdf-domain: DOMAIN
Authorization: <authorization_string>
```

```
DELETE /datatypes/<id>?domain=DOMAIN HTTP/1.1
Authorization: <authorization_string>
```

<id> is the UUID of the datatype to be deleted.

Request Parameters

This implementation of the operation does not use request parameters.

Request Headers

This implementation of the operation uses only the request headers that are common to most requests. See *Common Request Headers*.

Responses

Response Headers

This implementation of the operation uses only response headers that are common to most responses. See *Common Response Headers*.

Response Elements

This implementation of the operation does not return any response elements.

Special Errors

This implementation of the operation does not return special errors. For general information on standard error codes, see *Common Error Responses*.

Examples

Sample Request

```
DELETE /datatypes/t-6b0bdf9a-86b2-11e8-89f2-0242ac120009 HTTP/1.1
Host: hsdshdflab.hdfgroup.org
X-Hdf-domain: /shared/tall.h5
Content-Length: 0
Accept: */*
Accept-Encoding: gzip, deflate
```

Sample cURL command

```
$ curl -X DELETE -u username:password --header "X-Hdf-domain: /shared/tall.h5" \
↳ hsdshdflab.hdfgroup.org/datatypes/t-6b0bdf9a-86b2-11e8-89f2-0242ac120009
```

Sample Response

```
HTTP/1.1 200 OK
Date: Fri, 13 Jul 2018 15:49:44 GMT
Content-Type: application/json
Server: nginx/1.15.0
```

Related Resources

- *GET Attributes*
- *GET Datatype*
- *GET Datatypes*
- *POST Datatype*
- *POST Dataset*
- *PUT Attribute*

5.4.2 GET Datatype

Description

Returns information about the committed datatype with the UUID given in the URI.

Requests

Syntax

```
GET /datatypes/<id> HTTP/1.1
X-Hdf-domain: DOMAIN
Authorization: <authorization_string>
```

```
GET /datatypes/<id>?domain=DOMAIN HTTP/1.1
Authorization: <authorization_string>
```

<id> is the UUID of the requested datatype.

Request Parameters

This implementation of the operation does not use request parameters.

Request Headers

This implementation of the operation uses only the request headers that are common to most requests. See *Common Request Headers*.

Responses

Response Headers

This implementation of the operation uses only response headers that are common to most responses. See *Common Response Headers*.

Response Elements

On success, a JSON response will be returned with the following elements:

id

The UUID of the datatype object.

root

The root group of the domain which the datatype is within.

type

A JSON object representing the type of the datatype object.

attributeCount

The number of attributes belonging to the datatype.

created

A timestamp giving the time the datatype was created in UTC (ISO-8601 format).

lastModified

A timestamp giving the most recent time the datatype has been modified (i.e. attributes updated) in UTC (ISO-8601 format).

hrefs

An array of links to related resources. See *Hypermedia*.

Special Errors

This implementation of the operation does not return special errors. For general information on standard error codes, see *Common Error Responses*.

Examples

Get the committed datatype with UUID: "t-3e37ab7e-...".

Sample Request

```
GET /datatypes/t-3e37ab7e-86b3-11e8-bce3-0242ac12000c HTTP/1.1
Host: hsdshdflab.hdfgroup.org
X-Hdf-domain: /shared/tall.h5
Accept-Encoding: gzip, deflate
Accept: */*
```

Sample cURL command

```
$ curl -X GET --header "X-Hdf-domain: /shared/tall.h5" hsdshdflab.hdfgroup.org/
↳ datatypes/t-3e37ab7e-86b3-11e8-bce3-0242ac12000c
```

Sample Response

```
HTTP/1.1 200 OK
Date: Fri, 13 Jul 2018 15:57:37 GMT
Content-Length: 602
Etag: "c53bc5b2d3c3b5059b71ef92ca7d144a2df54456"
Content-Type: application/json
Server: nginx/1.15.0
```

```
{
  "id": "t-3e37ab7e-86b3-11e8-bce3-0242ac12000c",
  "root": "g-b116b6f0-85e9-11e8-9cc2-0242ac120008",
  "domain": "/shared/tall.h5",
  "type": {
    "class": "H5T_FLOAT",
    "base": "H5T_IEEE_F64LE"
  }
}
```

(continues on next page)

(continued from previous page)

```

    },
    "attributeCount": 0,
    "created": 1531496503.6064572,
    "lastModified": 1531496503.6064572,
    "hrefs": [
      {"rel": "self", "href": "hdsdshdflab.hdfgroup.org/datatypes/t-3e37ab7e-86b3-
↪11e8-bce3-0242ac12000c"},
      {"rel": "root", "href": "hdsdshdflab.hdfgroup.org/groups/g-b116b6f0-85e9-11e8-
↪9cc2-0242ac120008"},
      {"rel": "home", "href": "hdsdshdflab.hdfgroup.org/"},
      {"rel": "attributes", "href": "hdsdshdflab.hdfgroup.org/datatypes/t-3e37ab7e-
↪86b3-11e8-bce3-0242ac12000c/attributes"}
    ]
  }
}

```

Related Resources

- *DELETE Datatype*
- *GET Datatypes*
- *POST Datatype*
- *POST Dataset*
- *PUT Attribute*

5.4.3 GET Datatypes

Description

Gets all the committed datatypes in a domain.

Requests

Syntax

```

GET /datatypes HTTP/1.1
X-Hdf-domain: DOMAIN
Authorization: <authorization_string>

```

```

GET /datatypes?domain=DOMAIN HTTP/1.1
Authorization: <authorization_string>

```

Request Parameters

This implementation of the operation uses the following request parameters (both optional):

Limit

If provided, a positive integer value specifying the maximum number of UUIDs to return.

Marker

If provided, a string value indicating that only UUIDs that occur after the marker value will be returned.

Request Headers

This implementation of the operation uses only the request headers that are common to most requests. See *Common Request Headers*.

Responses

Response Headers

This implementation of the operation uses only response headers that are common to most responses. See *Common Response Headers*.

Response Elements

On success, a JSON response will be returned with the following elements:

hrefs

An array of links to related resources. See *Hypermedia*.

Special Errors

This implementation of the operation does not return special errors. For general information on standard error codes, see *Common Error Responses*.

Examples

Sample Request

```
GET /datatypes HTTP/1.1
Host: hsdshdflab.hdfgroup.org
X-Hdf-domain: /shared/tall.h5
Accept-Encoding: gzip, deflate
Accept: */*
```

Sample cURL command

```
$ curl -X GET --header "X-Hdf-domain: /shared/tall.h5" hsdshdfllab.hdfgroup.org/  
↳ datatypes
```

Sample Response

```
HTTP/1.1 200 OK  
Date: Fri, 13 Jul 2018 18:23:58 GMT  
Content-Length: 746  
Etag: "e01f56869a9a919b1496c463f3569a2a7c319f11"  
Content-Type: application/json  
Server: nginx/1.15.0
```

```
{  
  "datatypes": [  
    "t-3e37ab7e-86b3-11e8-bce3-0242ac12000c",  
    "t-af48f618-86c9-11e8-9cb8-0242ac120008",  
    "t-b0510adc-86c9-11e8-9361-0242ac120007",  
    "t-b0e855b8-86c9-11e8-ac22-0242ac12000d",  
    "t-b17d24fe-86c9-11e8-b364-0242ac12000a",  
    "t-b21dd3ea-86c9-11e8-9cb8-0242ac120008",  
    "t-b2b2a380-86c9-11e8-9361-0242ac120007",  
    "t-b34f3542-86c9-11e8-ac22-0242ac12000d",  
    "t-b4659570-86c9-11e8-b364-0242ac12000a",  
    "t-b4f32ffc-86c9-11e8-9cb8-0242ac120008",  
    "t-b58f2b50-86c9-11e8-9361-0242ac120007",  
    "t-b66113c2-86c9-11e8-ac22-0242ac12000d",  
    "t-b6f8e454-86c9-11e8-b364-0242ac12000a"  
  ],  
  "hrefs": [  
    {"href": "hsdshdfllab.hdfgroup.org/datatypes", "rel": "self"},  
    {"href": "hsdshdfllab.hdfgroup.org/groups/g-b116b6f0-85e9-11e8-9cc2-  
↳ 0242ac120008", "rel": "root"},  
    {"href": "hsdshdfllab.hdfgroup.org/", "rel": "home"}  
  ]  
}
```

Sample Request with Marker and Limit

This example uses the “Marker” request parameter to return only UUIDs after the given Marker value. Also, the “Limit” request parameter is used to limit the number of UUIDs in the response to 5.

```
GET /datatypes?Marker=t-b17d24fe-86c9-11e8-b364-0242ac12000a&Limit=5 HTTP/1.1  
Host: hsdshdfllab.hdfgroup.org  
X-Hdf-domain: /shared/tall.h5  
Accept-Encoding: gzip, deflate  
Accept: */*
```

Sample cURL command

URL enclosed in quotes to prevent shell from seeing ampersand

```
$ curl -X GET --header "X-Hdf-domain: /shared/tall.h5" "hdsdshdflab.hdfgroup.org/
↳datatypes?Marker=t-b17d24fe-86c9-11e8-b364-0242ac12000a&Limit=5"
```

Sample Response with Marker and Limit

```
HTTP/1.1 200 OK
Date: Fri, 13 Jul 2018 18:30:29 GMT
Content-Length: 410
Etag: "a2e2d5a3ae63cd504d02b51d99f27b30d17b75b5"
Content-Type: application/json
Server: nginx/1.15.0
```

```
{
  "datatypes": [
    "t-b21dd3ea-86c9-11e8-9cb8-0242ac120008",
    "t-b2b2a380-86c9-11e8-9361-0242ac120007",
    "t-b34f3542-86c9-11e8-ac22-0242ac12000d",
    "t-b4659570-86c9-11e8-b364-0242ac12000a",
    "t-b4f32ffc-86c9-11e8-9cb8-0242ac120008"
  ],
  "hrefs": [
    {"href": "hdsdshdflab.hdfgroup.org/datatypes", "rel": "self"},
    {"href": "hdsdshdflab.hdfgroup.org/groups/g-b116b6f0-85e9-11e8-9cc2-
↳0242ac120008", "rel": "root"},
    {"href": "hdsdshdflab.hdfgroup.org/", "rel": "home"}
  ]
}
```

Related Resources

- *DELETE Datatype*
- *GET Datatype*
- *POST Datatype*
- *POST Dataset*
- *PUT Attribute*

5.4.4 POST Datatype

Description

Creates a new committed datatype.

Requests

Syntax

```
POST /datatypes HTTP/1.1
X-Hdf-domain: DOMAIN
Authorization: <authorization_string>
```

```
POST /datatypes?domain=DOMAIN HTTP/1.1
Authorization: <authorization_string>
```

Request Parameters

This implementation of the operation does not use request parameters.

Request Headers

This implementation of the operation uses only the request headers that are common to most requests. See *Common Request Headers*.

Request Elements

The request body must be a JSON object with a 'type' link key as described below. Optionally, the request body can include a 'link' key that describes how the new committed datatype will be linked.

type

The value of the type key can either be one of the predefined type strings (see predefined types), or a JSON representation of a type. (see *Types*).

link

If present, the link value must include the following subkeys:

link['id']

The UUID of the group the new datatype should be linked from. If the UUID is not valid, the request will fail and a new datatype will not be created.

link['name']

The name of the new link.

Responses

Response Headers

This implementation of the operation uses only response headers that are common to most responses. See *Common Response Headers*.

Response Elements

On success, a JSON response will be returned with the following elements:

id

The UUID of the newly created datatype object.

root

The root group of the domain which the datatype is within.

attributeCount

The number of attributes belonging to the datatype.

created

A timestamp giving the time the datatype was created in UTC (ISO-8601 format).

lastModified

A timestamp giving the most recent time the datatype has been modified (i.e. attributes or links updated) in UTC (ISO-8601 format).

Special Errors

This implementation of the operation does not return special errors. For general information on standard error codes, see *Common Error Responses*.

Examples

Sample Request

Create a new committed datatype using the “H5T_IEEE_F32LE” (32-bit float) predefined type.

```
POST /datatypes HTTP/1.1
Host: hsdshdflab.hdfgroup.org
X-Hdf-domain: /shared/tall.h5
Content-Length: 26
Accept: */*
Accept-Encoding: gzip, deflate
```

```
{
  "type": "H5T_IEEE_F32LE"
}
```

Sample cURL command

```
$ curl -X POST -u username:password --header "X-Hdf-domain: /shared/tall.h5" --header
↳"Content-Type: application/json"
-d "{\"type\": \"H5T_IEEE_F32LE\"}" hsdshdflab.hdfgroup.org/datatypes
```

Sample Response

```
HTTP/1.1 201 Created
Date: Fri, 13 Jul 2018 15:35:49 GMT
Content-Length: 186
Content-Type: application/json
Server: nginx/1.15.0
```

```
{
  "id": "t-6b0bdf9a-86b2-11e8-89f2-0242ac120009",
  "created": 1531496149.3141127,
  "root": "g-b116b6f0-85e9-11e8-9cc2-0242ac120008",
  "lastModified": 1531496149.3141127,
  "attributeCount": 0
}
```

Sample Request with Link

Create a new committed datatype and link to root (id “g-b116b6f0-...”) as “linked_dtype”.

```
POST /datatypes HTTP/1.1
Host: hsdshdflab.hdfgroup.org
X-Hdf-domain: /shared/tall.h5
Content-Length: 108
Accept: */*
Accept-Encoding: gzip, deflate
```

```
{
  "type": "H5T_IEEE_F64LE",
  "link": {
    "id": "g-b116b6f0-85e9-11e8-9cc2-0242ac120008",
    "name": "linked_dtype"
  }
}
```

Sample cURL command

```
$ curl -X POST -u username:password --header "X-Hdf-domain: /shared/tall.h5" --header
↳"Content-Type: application/json"
-d "{\"type\": \"H5T_IEEE_F64LE\", \"link\": {\"id\": \"g-b116b6f0-85e9-11e8-9cc2-
↳0242ac120008\", \"name\": \"linked_dtype\"}}" hsdshdflab.hdfgroup.org/datatypes
```

Sample Response with Link

```
HTTP/1.1 201 Created
Date: Fri, 13 Jul 2018 15:41:44 GMT
Content-Length: 186
Content-Type: application/json
Server: nginx/1.15.0
```

```
{
  "id": "t-3e37ab7e-86b3-11e8-bce3-0242ac12000c",
  "root": "g-b116b6f0-85e9-11e8-9cc2-0242ac120008",
  "attributeCount": 0,
  "created": 1531496503.6064572,
  "lastModified": 1531496503.6064572
}
```

Related Resources

- *DELETE Datatype*
- *GET Datatype*
- *GET Datatypes*
- *POST Dataset*
- *PUT Attribute*

Like datasets (see *Datasets*), attributes are objects that contain a homogeneous collection of elements and have associated type information. Attributes are typically small metadata objects that describe some aspect of the object (dataset, group, or committed datatype) that contains the attribute.

6.1 Creating Attributes

Use *PUT Attribute* to create an attribute. If there is an existing attribute with the same name, it will be overwritten by this request. You can use *GET Attribute* to inquire if the attribute already exists or not. When creating an attribute, the attribute name, type and shape (for non-scalar attributes) is included in the request.

6.2 Reading and Writing Data

Unlike datasets, an attribute's data can not be read or written partially. Data can only be written as part of the PUT request. Reading the data of an attribute is done by *GET Attribute*.

6.3 Listing attributes

Use *GET Attributes* to get information about all the attributes of a group, dataset, or committed datatype.

6.4 Deleting Attributes

Use *DELETE Attribute* to delete an attribute.

6.5 List of Operations

6.5.1 DELETE Attribute

Description

The implementation of the DELETE operation deletes the attribute named in the URI.

Requests

Syntax

```
DELETE /groups/<id>/<name> HTTP/1.1
X-Hdf-domain: DOMAIN
Authorization: <authorization_string>
```

```
DELETE /groups/<id>/<name>?domain=DOMAIN HTTP/1.1
Authorization: <authorization_string>
```

```
DELETE /datasets/<id>/<name> HTTP/1.1
X-Hdf-domain: DOMAIN
Authorization: <authorization_string>
```

```
DELETE /datasets/<id>/<name>?domain=DOMAIN HTTP/1.1
Authorization: <authorization_string>
```

```
DELETE /datatypes/<id>/<name> HTTP/1.1
X-Hdf-domain: DOMAIN
Authorization: <authorization_string>
```

```
DELETE /datatypes/<id>/<name>?domain=DOMAIN HTTP/1.1
Authorization: <authorization_string>
```

- *<id>* is the UUID of the dataset/group/committed datatype
- *<name>* is the url-encoded name of the requested attribute

Request Parameters

This implementation of the operation does not use request parameters.

Request Headers

This implementation of the operation uses only the request headers that are common to most requests. See *Common Request Headers*

Responses

Response Headers

This implementation of the operation uses only response headers that are common to most responses. See *Common Response Headers*.

Response Elements

On success, a JSON response will be returned with the following elements:

hrefs

An array of links to related resources. See *Hypermedia*.

Special Errors

This implementation of the operation does not return special errors. For general information on standard error codes, see *Common Error Responses*.

Examples

Sample Request

```
DELETE /groups/g-45f464d8-883e-11e8-a9dc-0242ac12000e/attributes/attr1 HTTP/1.1
Host: hsdshdfllab.hdfgroup.org
X-Hdf-domain: /shared/tall.h5
Accept: */*
Accept-Encoding: gzip, deflate
```

Sample cURL command

```
$ curl -X DELETE -u username:password --header "X-Hdf-domain: /shared/tall.h5" \
  → hsdshdfllab.hdfgroup.org/groups/g-45f464d8-883e-11e8-a9dc-0242ac12000e/attributes/ \
  → attr1
```

Sample Response

```
HTTP/1.1 200 OK
Date: Sun, 15 Jul 2018 16:06:54 GMT
Content-Length: 13
Content-Type: application/json
Server: nginx/1.15.0
```

```
{"hrefs": []}
```

Related Resources

- *GET Attributes*
- *GET Attribute*
- *GET Dataset*
- *GET Datatype*
- *GET Group*
- *PUT Attribute*

6.5.2 GET Attribute

Description

Gets the specified attribute of a dataset, group, or committed datatype.

Requests

Syntax

To get an attribute of a group:

```
GET /groups/<id>/attributes/<name> HTTP/1.1
X-Hdf-domain: DOMAIN
Authorization: <authorization_string>
```

```
GET /groups/<id>/attributes/<name>?domain=DOMAIN HTTP/1.1
Authorization: <authorization_string>
```

To get an attribute of a dataset:

```
GET /datasets/<id>/attributes/<name> HTTP/1.1
X-Hdf-domain: DOMAIN
Authorization: <authorization_string>
```

```
GET /datasets/<id>/attributes/<name>?domain=DOMAIN HTTP/1.1
Authorization: <authorization_string>
```

To get an attribute of a datatype:

```
GET /datatypes/<id>/attributes/<name> HTTP/1.1
X-Hdf-domain: DOMAIN
Authorization: <authorization_string>
```

```
GET /datatypes/<id>/attributes/<name>?domain=DOMAIN HTTP/1.1
Authorization: <authorization_string>
```

where:

- *<id>* is the UUID of the dataset/group/committed datatype
- *<name>* is the url-encoded name of the requested attribute

Request Parameters

This implementation of the operation does not use request parameters.

Request Headers

This implementation of the operation uses only the request headers that are common to most requests. See *Common Request Headers*.

Responses

Response Headers

This implementation of the operation uses only response headers that are common to most responses. See *Common Response Headers*.

Response Elements

On success, a JSON response will be returned with the following elements:

type

A JSON object representing the type of the attribute. See *Types* for details of the type representation.

shape

A JSON object that represents the dataspace of the attribute. Subkeys of shape are:

class: A string with one of the following values:

- H5S_NULL: A null dataspace, which has no elements
- H5S_SCALAR: A dataspace with a single element (although possibly of a complex datatype)
- H5S_SIMPLE: A dataspace that consists of a regular array of elements

dims: An integer array whose length is equal to the number of dimensions (rank) of the dataspace. The value of each element gives the current size of each dimension. Dims is not returned for H5S_NULL or H5S_SCALAR dataspaces.

value

A JSON array (or string or number for scalar datasets) giving the values of the requested attribute.

hrefs

An array of links to related resources. See *Hypermedia*.

Special Errors

This implementation of the operation does not return special errors. For general information on standard error codes, see *Common Error Responses*.

Examples

Sample Request

Get an attribute named “attr1” from a group with UUID: “g-be5996fa-...”.

```
GET /groups/g-be5996fa-83c5-11e8-a8e6-0242ac120016/attributes/attr1 HTTP/1.1
Host: hsdshdflab.hdfgroup.org
X-Hdf-domain: /shared/tall.h5
Accept-Encoding: gzip, deflate
Accept: */*
```

Sample cURL command

```
$ curl -X GET --header "X-Hdf-domain: /shared/tall.h5" hsdshdflab.hdfgroup.org/groups/
↪g-be5996fa-83c5-11e8-a8e6-0242ac120016/attributes/attr1
```

Sample Response

```
HTTP/1.1 200 OK
Date: Sun, 15 Jul 2018 16:17:44 GMT
Content-Length: 522
Etag: "55b2e2ce2d3a2449a49cfd76c4dae635ec43a150"
Content-Type: application/json
Server: nginx/1.15.0
```

```
{
  "name": "attr1",
  "type": {
    "base": "H5T_STD_I8LE",
    "class": "H5T_INTEGER"
  },
  "shape": {
    "class": "H5S_SIMPLE",
    "dims": [10]
  },
  "created": 1531174596.117736,
  "lastModified": 1531174596.117736,
  "value": [97, 98, 99, 100, 101, 102, 103, 104, 105, 0],
  "hrefs": [
    {"href": "hsdshdflab.hdfgroup.org/groups/g-be5996fa-83c5-11e8-a8e6-
↪0242ac120016/attributes/attr1", "rel": "self"},
    {"href": "hsdshdflab.hdfgroup.org/", "rel": "home"},
    {"href": "hsdshdflab.hdfgroup.org/groups/g-be5996fa-83c5-11e8-a8e6-
↪0242ac120016", "rel": "owner"}
  ]
}
```

Related Resources

- *DELETE Attribute*
- *GET Attributes*
- *GET Dataset*
- *GET Datatype*
- *GET Group*
- *PUT Attribute*

6.5.3 GET Attributes

Description

Gets all the attributes of a dataset, group, or committed datatype. For each attribute the request returns the attribute's name, type, and shape. To get the attribute data use *GET Attribute*.

Requests

Syntax

To get the attributes of a group:

```
GET /groups/<id>/attributes HTTP/1.1
X-Hdf-domain: DOMAIN
Authorization: <authorization_string>
```

```
GET /groups/<id>/attributes?domain=DOMAIN HTTP/1.1
Authorization: <authorization_string>
```

To get the attributes of a dataset:

```
GET /datasets/<id>/attributes HTTP/1.1
X-Hdf-domain: DOMAIN
Authorization: <authorization_string>
```

```
GET /datasets/<id>/attributes?domain=DOMAIN HTTP/1.1
Authorization: <authorization_string>
```

To get the attributes of a datatype:

```
GET /datatypes/<id>/attributes HTTP/1.1
X-Hdf-domain: DOMAIN
Authorization: <authorization_string>
```

```
GET /datatypes/<id>/attributes?domain=DOMAIN HTTP/1.1
Authorization: <authorization_string>
```

where:

- *<id>* is the UUID of the dataset/group/committed datatype

Request Parameters

This implementation of the operation uses the following request parameters (both optional):

Limit

If provided, a positive integer value specifying the maximum number of attributes to return.

Marker

If provided, a string value indicating that only attributes that occur after the marker value will be returned. *Note:* the marker expression should be url-encoded.

Request Headers

This implementation of the operation uses only the request headers that are common to most requests. See *Common Request Headers*.

Responses

Response Headers

This implementation of the operation uses only response headers that are common to most responses. See *Common Response Headers*.

Response Elements

On success, a JSON response will be returned with the following elements:

attributes

An array of JSON objects with an element for each returned attribute. Each element will have keys: name, type, shape, created, and lastModified. See *GET Attribute* for a description of these keys.

hrefs

An array of links to related resources. See *Hypermedia*.

Special Errors

This implementation of the operation does not return special errors. For general information on standard error codes, see *Common Error Responses*.

Examples

Sample Request

Get attributes of a group with UUID: "1a956e54-...".

```
GET /groups/g-be5996fa-83c5-11e8-a8e6-0242ac120016/attributes HTTP/1.1
Host: hsdshdflab.hdfgroup.org
X-Hdf-domain: /shared/tall.h5
Accept-Encoding: gzip, deflate
Accept: */*
```

Sample cURL command

```
$ curl -X GET --header "X-Hdf-domain: /shared/tall.h5" hsdshdflab.hdfgroup.org/groups/
↪g-be5996fa-83c5-11e8-a8e6-0242ac120016/attributes
```

Sample Response

```
HTTP/1.1 200 OK
Date: Sun, 15 Jul 2018 16:23:43 GMT
Content-Length: 797
Etag: "7cbeefcf8d9997a8865bdea3bf2d541a14e9bf71"
Content-Type: application/json
Server: nginx/1.15.0
```

```
{
  "attributes": [
    {
      "type": {
        "base": "H5T_STD_I8LE",
        "class": "H5T_INTEGER"
      },
      "name": "attr1",
      "shape": {
        "dims": [10],
        "class": "H5S_SIMPLE"
      },
      "created": 1531174596.117736,
      "href": "hsdshdflab.hdfgroup.org/groups/g-be5996fa-83c5-11e8-a8e6-
↪0242ac120016/attributes/attr1"
    },
    {
      "type": {
        "base": "H5T_STD_I32BE",
        "class": "H5T_INTEGER"
      },
      "name": "attr2",
      "shape": {
        "dims": [2, 2],
        "class": "H5S_SIMPLE"
      }
    }
  ]
}
```

(continues on next page)

(continued from previous page)

```

        "created": 1531174596.141592,
        "href": "hdsdshdflab.hdfgroup.org/groups/g-be5996fa-83c5-11e8-a8e6-
↪0242ac120016/attributes/attr2"
    },
    ],
    "hrefs": [
        {"href": "hdsdshdflab.hdfgroup.org/groups/g-be5996fa-83c5-11e8-a8e6-
↪0242ac120016/attributes", "rel": "self"},
        {"href": "hdsdshdflab.hdfgroup.org/", "rel": "home"},
        {"href": "hdsdshdflab.hdfgroup.org/groups/g-be5996fa-83c5-11e8-a8e6-
↪0242ac120016", "rel": "owner"}
    ]
}

```

Sample Request - get Batch

Get the five attributes that occur after attribute “attr2” from a group with UUID: “g-45f464d8-...”.

```

GET /groups/g-45f464d8-883e-11e8-a9dc-0242ac12000e/attributes?Marker=attr2&Limit=5_
↪HTTP/1.1
Host: hdsdshdflab.hdfgroup.org
X-Hdf-domain: /shared/tall.h5
Accept-Encoding: gzip, deflate
Accept: */*

```

Sample cURL command

```

$ curl -X GET --header "X-Hdf-domain: /shared/tall.h5" "hdsdshdflab.hdfgroup.org/
↪groups/g-45f464d8-883e-11e8-a9dc-0242ac12000e/attributes?Marker=attr2&Limit=5"

```

Sample Response - get Batch

```

HTTP/1.1 200 OK
Date: Sun, 15 Jul 2018 16:38:16 GMT
Content-Length: 1767
Etag: "9483f4356e08d12b719aa64ece09e659b05adaf2"
Content-Type: application/json
Server: nginx/1.15.0

```

```

{
  "attributes": [
    {
      "name": "attr3",
      "type": {
        "base": "H5T_STD_U32BE",
        "class": "H5T_INTEGER"
      },
      "shape": {
        "class": "H5S_SCALAR"
      }
    },
    ...
  ]
}

```

(continues on next page)

(continued from previous page)

```

      "created": 1531672545.5978162,
      "href": "hdsdshdflab.hdfgroup.org/groups/g-45f464d8-883e-11e8-a9dc-
↪0242ac12000e/attributes/attr3"
    },
    {
      "name": "attr4",
      "type": {
        "base": "H5T_STD_I32LE",
        "class": "H5T_INTEGER"
      },
      "shape": {
        "class": "H5S_SCALAR"
      },
      "created": 1531667223.0914037,
      "href": "hdsdshdflab.hdfgroup.org/groups/g-45f464d8-883e-11e8-a9dc-
↪0242ac12000e/attributes/attr4"
    },
    {
      "name": "attr5",
      "type": {
        "base": "H5T_STD_U64LE",
        "class": "H5T_INTEGER"
      },
      "shape": {
        "class": "H5S_SCALAR"
      },
      "created": 1531672562.6758137,
      "href": "hdsdshdflab.hdfgroup.org/groups/g-45f464d8-883e-11e8-a9dc-
↪0242ac12000e/attributes/attr5"
    },
    {
      "name": "attr6",
      "type": {
        "strPad": "H5T_STR_NULLPAD",
        "charSet": "H5T_CSET_ASCII",
        "class": "H5T_STRING",
        "length": 40
      },
      "shape": {
        "class": "H5S_SIMPLE",
        "dims": [2]
      },
      "created": 1531668943.5116098,
      "href": "hdsdshdflab.hdfgroup.org/groups/g-45f464d8-883e-11e8-a9dc-
↪0242ac12000e/attributes/attr6"
    },
    {
      "name": "attr7",
      "type": {
        "base": "H5T_STD_U64LE",
        "class": "H5T_INTEGER"
      },
      "shape": {
        "class": "H5S_SCALAR"
      },
      "created": 1531672573.915442,
      "href": "hdsdshdflab.hdfgroup.org/groups/g-45f464d8-883e-11e8-a9dc-
↪0242ac12000e/attributes/attr7"

```

(continues on next page)

(continued from previous page)

```

    },
    "hrefs": [
      {"href": "hdsdshdf1ab.hdfgroup.org/groups/g-45f464d8-883e-11e8-a9dc-
↪0242ac12000e/attributes", "rel": "self"},
      {"href": "hdsdshdf1ab.hdfgroup.org/", "rel": "home"},
      {"href": "hdsdshdf1ab.hdfgroup.org/groups/g-45f464d8-883e-11e8-a9dc-
↪0242ac12000e", "rel": "owner"}
    ]
  }
}

```

Related Resources

- *DELETE Attribute*
- *GET Attributes*
- *GET Dataset*
- *GET Datatype*
- *GET Group*
- *PUT Attribute*

6.5.4 PUT Attribute

Description

Creates a new attribute in a group, dataset, or committed datatype.

Note: The new attribute will replace any existing attribute with the same name.

Requests

Syntax

To create a group attribute:

```

PUT /groups/<id>/attributes/<name> HTTP/1.1
X-Hdf-domain: DOMAIN
Authorization: <authorization_string>

```

```

PUT /groups/<id>/attributes/<name>?domain=DOMAIN HTTP/1.1
Authorization: <authorization_string>

```

To create a dataset attribute:

```

PUT /datasets/<id>/attributes/<name> HTTP/1.1
X-Hdf-domain: DOMAIN
Authorization: <authorization_string>

```

```

PUT /datasets/<id>/attributes/<name>?domain=DOMAIN HTTP/1.1
Authorization: <authorization_string>

```

To create a committed datatype attribute:

```
PUT /datatypes/<id>/attributes/<name> HTTP/1.1
X-Hdf-domain: DOMAIN
Authorization: <authorization_string>
```

```
PUT /datatypes/<id>/attributes/<name>?domain=DOMAIN HTTP/1.1
Authorization: <authorization_string>
```

- *<id>* is the UUID of the dataset/group/committed datatype
- *<name>* is the url-encoded name of the requested attribute

Request Parameters

This implementation of the operation does not use request parameters.

Request Headers

This implementation of the operation uses only the request headers that are common to most requests. See *Common Request Headers*

Request Elements

The request body must include a JSON object with “type” key. Optionally a “shape” key can be provided to make a non-scalar attribute.

type

Specifies the desired type of the attribute. Either a string that is one of the predefined type values, a UUID of a committed type, or a JSON object describing the type. See *Types* for details of the type specification.

shape

Either a string with the value `H5S_NULL` or an integer array describing the dimensions of the attribute. If shape is not provided, a scalar attribute will be created. If a shape value of `H5S_NULL` is specified a null space attribute will be created. (Null space attributes can not contain any data values.)

value

A JSON array (or number or string for scalar attributes with primitive types) that specifies the initial values for the attribute. The elements of the array must be compatible with the type of the attribute. Not valid to provide if the shape is `H5S_NULL`.

Responses

Response Headers

This implementation of the operation uses only response headers that are common to most responses. See *Common Response Headers*.

Response Elements

On success, a JSON response will be returned with the following elements:

hrefs

An array of links to related resources. See *Hypermedia*.

Special Errors

This implementation of the operation does not return special errors. For general information on standard error codes, see *Common Error Responses*.

Examples

Sample Request - scalar attribute

Create an integer scalar attribute in the group with UUID of “g-45f464d8-” named “attr4”. The value of the attribute will be 42.

```
PUT /groups/g-45f464d8-883e-11e8-a9dc-0242ac12000e/attributes/attr4 HTTP/1.1
Host: hsdshdflab.hdfgroup.org
X-Hdf-domain: /shared/tall.h5
Content-Length: 38
Accept: */*
Accept-Encoding: gzip, deflate
```

```
{
  "type": "H5T_STD_I32LE",
  "value": 42
}
```

Sample cURL command

```
$ curl -X PUT -u username:password --header "X-Hdf-domain: /shared/tall.h5" --header
↪ "Content-Type: application/json"
  -d "{\"type\": \"H5T_STD_I32LE\", \"value\": 42}" hsdshdflab.hdfgroup.org/groups/g-
↪ 45f464d8-883e-11e8-a9dc-0242ac12000e/attributes/attr4
```

Sample Response - scalar attribute

```
HTTP/1.1 201 Created
Date: Sun, 15 Jul 2018 15:07:03 GMT
Content-Length: 13
Content-Type: application/json
Server: nginx/1.15.0
```

```
{"hrefs": []}
```

Sample Request - string attribute

Create a two-element, fixed width string attribute in the group with UUID of “g-45f464d8-” named “attr6”. The attributes values will be “Hello,…” and “Goodbye!”.

```
PUT /groups/g-45f464d8-883e-11e8-a9dc-0242ac12000e/attributes/attr6 HTTP/1.1
Host: hsdshdfllab.hdfgroup.org
X-Hdf-domain: /shared/tall.h5
Content-Length: 178
Accept: */*
Accept-Encoding: gzip, deflate
```

```
{
  "shape": [2],
  "type": {
    "class": "H5T_STRING",
    "cset": "H5T_CSET_ASCII",
    "strpad": "H5T_STR_NULLPAD",
    "strsize": 40
  },
  "value": ["Hello, I'm a fixed-width string!", "Goodbye!"]
}
```

Sample cURL command

```
$ curl -X PUT -u username:password --header "X-Hdf-domain: /shared/tall.h5" --header
↪ "Content-Type: application/json"
  -d '{"shape": [2], "type": {"class": "H5T_STRING", "charSet": "H5T_CSET_
↪ ASCII", "strPad": "H5T_STR_NULLPAD", "length": 40},
  "value": ["Hello, I'm a fixed-width string"!'!', "Goodbye"!'!']}'
↪ hsdshdfllab.hdfgroup.org/groups/g-45f464d8-883e-11e8-a9dc-0242ac12000e/attributes/
↪ attr6
```

Sample Response - string attribute

```
HTTP/1.1 201 Created
Date: Sun, 15 Jul 2018 15:35:43 GMT
Content-Length: 13
Content-Type: application/json
Server: nginx/1.15.0
```

```
{"hrefs": []}
```

Sample Request - compound type

Create a two-element, attribute in the group with UUID of “g-45f464d8-” named “attr_compound”. The attribute has a compound type with an integer and a floating point element.

```
PUT /groups/g-45f464d8-883e-11e8-a9dc-0242ac12000e/attributes/attr_compound HTTP/1.1
Host: hsdshdfllab.hdfgroup.org
X-Hdf-domain: /shared/tall.h5
Content-Length: 187
Accept: */*
Accept-Encoding: gzip, deflate
```

```
{
  "shape": 2,
  "type": {
    "class": "H5T_COMPOUND",
    "fields": [
      {"type": "H5T_STD_I32LE", "name": "temp"},
      {"type": "H5T_IEEE_F32LE", "name": "pressure"}
    ]
  },
  "value": [[55, 32.34], [59, 29.34]]
}
```

Sample cURL command

```
$ curl -X PUT -u username:password --header "X-Hdf-domain: /shared/tall.h5" --header
↪ "Content-Type: application/json"
  -d '{"shape": 2, "type": {"class": "H5T_COMPOUND", "fields": [{"type": \
↪ "H5T_STD_I32LE", "name": "temp"},
  {"type": "H5T_IEEE_F32LE", "name": "pressure"}]}, "value": [[55, 32.34],
↪ [59, 29.34]]}' hsdshdfllab.hdfgroup.org/groups/g-45f464d8-883e-11e8-a9dc-
↪ 0242ac12000e/attributes/attr_compound
```

Sample Response - compound type

```
HTTP/1.1 201 Created
Date: Sun, 15 Jul 2018 15:43:00 GMT
Content-Length: 13
Content-Type: application/json
Server: nginx/1.15.0
```

```
{"hrefs": []}
```

Related Resources

- *DELETE Attribute*

- *GET Attribute*
- *GET Attributes*
- *GET Dataset*
- *GET Datatype*
- *GET Group*

The HDF REST API supports the rich type capabilities provided by HDF. Types are described in JSON and these JSON descriptions are used in operations involving datasets, attributes, and committed types.

There is not a separate request for creating types; rather, the description of the type is included with the request to create the dataset, attribute, or committed type. Once a type is created it is immutable and will exist until the containing object is deleted.

Type information is returned as a JSON object in dataset, attribute, or committed type GET requests (under the type key).

7.1 Predefined Types

Predefined types are base integer and floating point types that are identified via one of the following strings:

- `H5T_STD_U8 {LE | BE}`: a one byte unsigned integer
- `H5T_STD_I8 {LE | BE}`: a one byte signed integer
- `H5T_STD_U16 {LE | BE}`: a two byte unsigned integer
- `H5T_STD_I16 {LE | BE}`: a two byte signed integer
- `H5T_STD_U32 {LE | BE}`: a four byte unsigned integer
- `H5T_STD_I32 {LE | BE}`: a four byte signed integer
- `H5T_STD_U64 {LE | BE}`: an eight byte unsigned integer
- `H5T_STD_I64 {LE | BE}`: an eight byte signed integer
- `H5T_IEEE_F32 {LE | BE}`: a four byte floating-point value
- `H5T_IEEE_F64 {LE | BE}`: an eight byte floating-point value

Predefined types ending in “LE” are little-endian formatted and types ending in “BE” are big-endian. E.g. `H5T_STD_I64LE` would be an eight byte, signed, little-endian integer.

Note: “little vs. big endian” is used to specify the byte ordering in the server storage system and is not reflected in the JSON representation of the values.

7.1.1 Example

JSON Representation of an attribute with a H5T_STD_I8LE (signed, one byte) type:

```
{
  "name": "attr1",
  "shape": {
    "class": "H5S_SIMPLE",
    "dims": [27]
  },
  "type": {
    "class": "H5T_INTEGER",
    "base": "H5T_STD_I8LE"
  },
  "value": [49, 115, 116, 32, 97, 116, 116, 114, 105, 98, 117, 116, 101, 32,
           111, 102, 32, 100, 115, 101, 116, 49, 46, 49, 46, 49, 0]
}
```

7.2 String Types - Fixed Length

Fixed length strings have a specified length (supplied when the object is created) that is used for each data element. Any values that are assigned that exceed that length will be truncated.

To specify a fixed length string, create a JSON object with class, charSet, strPad, and length keys (see definitions of these keys below).

Note: Current only the ASCII character set is supported.

Note: Fixed width unicode strings are not currently supported.

Note: String Padding other than “H5T_STR_NULLPAD” will get stored as “H5T_STR_NULLPAD”

7.2.1 Example

JSON representation of a dataset using a fixed width string of 40 characters:

```
{
  "id": "d-be8bace4-83c5-11e8-90e7-0242ac120013",
  "shape": {
    "class": "H5S_SCALAR",
  },
  "type": {
    "class": "H5T_STRING",
    "charSet": "H5T_CSET_ASCII",
    "strPad": "H5T_STR_NULLPAD",
    "length": 40
  },
  "value": "Hello, World!"
}
```

7.3 String Types - Variable Length

Variable length strings allow each element of an array to only use as much storage as needed. This is convenient when the maximum string length is not known before hand, or there is a great deal of variability in the lengths of strings.

Note: Typically there is a slight performance penalty in accessing variable length string elements of an array in the server.

To specify a variable length string, create a JSON object with class, charSet, strPad, and length keys (see definitions of these keys below) where the value of “length” is: H5T_VARIABLE.

Note: Current only the ASCII character set is supported.

Note: Variable width unicode strings are not currently supported.

Note: String Padding other than “H5T_STR_NULLTERM” will get stored as “H5T_STR_NULLTERM”

7.3.1 Example

JSON representation of a attribute using a variable length string:

```
{
  "name": "A1",
  "shape": {
    "class": "H5S_SIMPLE",
    "dims": [4]
  },
  "type": {
    "class": "H5T_STRING",
    "charSet": "H5T_CSET_ASCII",
    "strPad": "H5T_STR_NULLTERM",
    "length": "H5T_VARIABLE"
  },
  "value": [
    "Hypermedia",
    "as the",
    "engine",
    "of state."
  ]
}
```

7.4 Compound Types

For some types of data it makes sense to store sets of related items together rather than in separate datasets or attributes. For these use cases a compound datatype can be defined. A compound datatype has class: H5T_COMPOUND and a field key which contains an array of sub-types. Each of these sub-types can be a primitive type, a string, or another compound type. Each sub-type has a name that can be used to refer to the element.

Note: The field names are not shown in the representation of a dataset’s or attribute’s values.

7.4.1 Example

JSON representation of a scalar attribute with a compound type that consists of two floating point elements:

```

{
  "name": "mycomplex",
  "shape": {
    "class": "H5S_SCALAR"
  },
  "type": {
    "class": "H5T_COMPOUND",
    "fields": [
      {
        "name": "real_part",
        "type": {
          "base": "H5T_IEEE_F64LE",
          "class": "H5T_FLOAT"
        }
      },
      {
        "name": "imaginary_part",
        "type": {
          "base": "H5T_IEEE_F64LE",
          "class": "H5T_FLOAT"
        }
      }
    ]
  },
  "value": [ 1.2345, -2.468 ]
}

```

7.5 Enumerated Types

Enumerated types enable the integer values of a dataset or attribute to be mapped to a set of strings. This allows the semantic meaning of a given set of values to be described along with the data.

To specify an enumerated type, use the class `H5T_ENUM`, provide a base type (must be some form of integer), and a “mapping” key that lists strings with their associated numeric values.

7.5.1 Example

```

{
  "id": "d-be9c3582-83c5-11e8-947e-0242ac120014",
  "shape": {
    "class": "H5S_SIMPLE",
    "dims": [ 7 ]
  },
  "type": {
    "class": "H5T_ENUM",
    "base": {
      "class": "H5T_INTEGER",
      "base": "H5T_STD_I16BE"
    },
    "mapping": {
      "GAS": 2,
      "LIQUID": 1,
      "PLASMA": 3,

```

(continues on next page)

(continued from previous page)

```

        "SOLID": 0
      },
      "value": [ 0, 2, 3, 2, 0, 1, 1 ]
    }
  }
}

```

7.6 Array Types

Array types are used when it is desired for each element of an attribute or dataset to itself be a (typically small) array.

To specify an array type, use the class `H5T_ARRAY` and provide the dimensions of the array with the type. Use the “base” key to specify the type of the elements of the array type.

7.6.1 Example

A dataset with 3 elements, each of which is a 2x2 array of integers.

```

{
  "id": "d-bf1cb98c-83c5-11e8-b9ee-0242ac12000a",
  "shape": {
    "class": "H5S_SIMPLE",
    "dims": [ 3 ]
  },
  "type": {
    "class": "H5T_ARRAY",
    "base": {
      "class": "H5T_INTEGER",
      "base": "H5T_STD_I16BE"
    },
    "dims": [ 2, 2 ]
  },
  "value": [
    [ [1, 2], [3, 4] ],
    [ [2, 1], [4, 3] ],
    [ [1, 1], [4, 4] ]
  ]
}

```

7.7 Opaque Types

TBD

7.7.1 Example

TBD

7.8 Object Reference Types

An object reference type enables you to define an array where each element of the array is a reference to another dataset, group or committed datatype.

To specify an object reference type, use `H5T_REFERENCE` as the type class, and `H5T_STD_REF_OBJ` as the base type.

The elements of the array consist of strings that have the prefix: “groups/”, “datasets/”, or “datatypes/” followed by the UUID of the referenced object.

7.8.1 Example

A JSON representation of an attribute that consists of a 3 element array of object references. The first element points to a group, the second element is null, and the third element points to a group.

```
{
  "name": "objref_attr",
  "shape": {
    "class": "H5S_SIMPLE",
    "dims": [ 3 ]
  },
  "type": {
    "class": "H5T_REFERENCE",
    "base": "H5T_STD_REF_OBJ"
  },
  "value": [
    "groups/g-be836c0a-83c5-11e8-947e-0242ac120014",
    "",
    "datasets/d-be8bace4-83c5-11e8-90e7-0242ac120013"
  ]
}
```

7.9 Region Reference Types

A region reference type allows the creation of attributes or datasets where each array element references a section (point selection or hyperslab) of another dataset.

To specify a region reference type, use `H5T_REFERENCE` as the type class, and `H5T_STD_REF_DSETREG` as the base type.

Note: When writing values to the dataset, each element of the dataset must be a JSON object with keys: “id”, “select_type”, and “selection” (as in the example below).

7.9.1 Example

A JSON representation of a region reference dataset with two elements.

The first element is a point selection element that references 4 elements in the dataset with UUID of “d-be9c3582-...”.

The second element is a hyperslab selection that references 4 hyper-slabs in the same dataset as the first element. Each element is a pair of points that gives the boundary of the selection.

```

{
  "id": "d-bf1cb98c-83c5-11e8-b9ee-0242ac12000a",
  "shape": {
    "class": "H5S_SIMPLE",
    "dims": [2]
  },
  "type": {
    "class": "H5T_REFERENCE",
    "base": "H5T_STD_REF_DSETREG"
  },
  "value": [
    {
      "id": "d-be9c3582-83c5-11e8-947e-0242ac120014",
      "select_type": "H5S_SEL_POINTS",
      "selection": [
        [0, 1], [2, 11], [1, 0], [2, 4]
      ]
    },
    {
      "id": "d-be9c3582-83c5-11e8-947e-0242ac120014",
      "select_type": "H5S_SEL_HYPERSLABS",
      "selection": [
        [ [0, 0], [0, 2] ],
        [ [0, 11], [0, 13] ],
        [ [2, 0], [2, 2] ],
        [ [2, 11], [2, 13] ]
      ]
    }
  ]
}

```

7.10 Type Keys

Information on the JSON keys used in type specifications.

7.10.1 class

The type class. One of:

- H5T_INTEGER: an integer type
- H5T_FLOAT: a floating point type
- H5T_STRING: a string type
- H5T_OPAQUE: an opaque type
- H5T_COMPOUND: a compound type
- H5T_ARRAY: an array type
- H5T_ENUM: an enum type
- H5T_REFERENCE: a reference type

7.10.2 base

A string that gives the base predefined type used (or reference type for the reference class).

7.10.3 order

The byte ordering. One of:

- `H5T_NONE`: Ordering is not relevant (e.g. for string types)
- `H5T_ORDER_LE`: Little endian ordering (e.g. native ordering for x86 computers)
- `H5T_ORDER_BE`: Big endian ordering

7.10.4 charSet

Character set for strings. Currently only `H5T_CSET_ASCII` is supported.

7.10.5 strPad

Defines how fixed length strings are padded. One of:

- `H5T_STR_NULLPAD`: String is padded with nulls
- `H5T_STR_NULLTERM`: String is null terminated
- `H5T_STR_SPACEPAD`: String is padded with spaces

7.10.6 length

Defines the string length. Either a positive integer or the string: `H5T_VARIABLE`.

7.10.7 name

The field name for compound types.

7.10.8 mapping

The enum name for enum types.

7.10.9 select_type

The selection type for reference types. One of:

- `H5S_SEL_POINTS`: selection is a series of points
- `H5S_SEL_HYPERSLABS`: selection is a series of hyper-slabs.

7.11 Related Resources

- *GET Dataset*
- *GET Type*
- *POST Dataset*
- *GET Attribute*
- *PUT Attribute*
- *GET Datatype*
- *POST Datatype*

Access Control List

An Access Control List (ACL) is a key-value store that can be used to manage what operations can be performed by which user on group, dataset, or committed type objects. Operations on other objects (e.g. links, dataspace, or attributes) use the ACL of the object they belong to.

Each ACL consists of 1 or more items in the form:

(username, read, create, update, delete, readACL, updateACL)

where username is a string, and read, create, update, delete, readACL and updateACL are booleans. These flags have the following semantics when the given username is provided in the http Authorization header:

- read: The given user is authorized for read access to the resource (generally all GET requests)
- create: The given user is authorized to create new resources (generally POST or PUT requests)
- update: The given user is authorized to modified a resource (e.g. *PUT Value*)
- delete: The given user is authorized to delete a resource (e.g. Delete a Group)
- readACL: The given user is authorized to read the ACLs of a resource
- updateACL: The given user is authorized to modify the ACLs of a resource

A special username ‘default’ is used to denote the access permission for all other users who are not listed in the ACL (including un-authenticated requests that don’t provide a username).

8.1 Example

Suppose a given dataset has the following ACL:

username	read	create	update	delete	readACL	writeACL
default	true	false	false	false	false	false
joe	true	false	true	false	false	false
ann	true	true	true	true	true	true

This ACL would enable anyone to read (perform GET requests). User 'joe' would be able to read and update (modify values in the dataset). While user 'ann' would have full control to do any operation on the dataset (including modifying permissions for herself or other users).

The following unauthenticated (no HTTP Authorization header) requests on the dataset would be granted or denied as follows:

- GET /datasets/<id> - granted (returns HTTP Status 200 - OK)
- POST /datasets/<id>/value - granted (returns HTTP Status 200 - OK)
- PUT /datasets/<id>/shape) - denied (returns HTTP Status 401 - Unauthorized)
- PUT /datasets/<id>/attributes/<name> - denied (returns HTTP Status 401 - Unauthorized)
- DELETE /datasets/<id> - denied (returns HTTP Status 401 - Unauthorized)

Next the same set of requests are sent with 'joe' as the user in the HTTP Authorization header:

- GET /datasets/<id> - granted (returns HTTP Status 200 - OK)
- POST /datasets/<id>/value - granted (returns HTTP Status 200 - OK)
- PUT /datasets/<id>/shape) - grant (returns HTTP Status 200 - OK)
- PUT /datasets/<id>/attributes/<name> - denied (returns HTTP Status 403 - Forbidden)
- DELETE /datasets/<id> - denied (returns HTTP Status 403 - Forbidden)

Finally the same set of requests are sent with 'ann' as the user:

- GET /datasets/<id> - granted (returns HTTP Status 200 - OK)
- POST /datasets/<id>/value - granted (returns HTTP Status 200 - OK)
- PUT /datasets/<id>/shape) - grant (returns HTTP Status 200 - OK)
- PUT /datasets/<id>/attributes/<name> - denied (returns HTTP Status 201 - Created)
- DELETE /datasets/<id> - denied (returns HTTP Status 200 - OK)

Note: HTTP Status 401 basically says: "you can't have access until you tell me who you are", while HTTP Status 403 says: "I know who you are, but you don't have permissions to access this resource."

8.2 Root ACL Inheritance

In many cases it will be desired to have a default ACL that applies to each resource in the domain. This can be accomplished by defining an ACL for the root group. This will control the access rights for any resource unless an ACL is present in that resource for the requesting user.

The default ACL can be read or updated by forming a request with a uri that includes the root group id, i.e.: "/groups/<root_id>/acls", or by using the uri path for the domain, i.e. "/acls".

For a given user then, the permissions for a resource are found in the following way:

1. If the user is present in the resources ACL, those permissions are used
2. If no user is present in the resources ACL, but is present in the root group, those permissions are used
3. Otherwise, if a 'default' user is present in the resource ACL, those permissions are used
4. If a 'default' user is not present in the resource ACL, but is present in the root ACL, those permissions are used
5. If no 'default' user is present in the root ACL, the permissions defined in the 'default_acl' config is used

8.3 List of Operations

8.3.1 GET ACL

Description

Returns access information for the given user for the object with the UUID provided in the URI.

Requests

Syntax

To get a user's default access for a domain:

```
GET /acls/<userid> HTTP/1.1
Host: DOMAIN
Authorization: <authorization_string>
```

To get a user's access information for a group:

```
GET /groups/<id>/acls/<userid> HTTP/1.1
Host: DOMAIN
Authorization: <authorization_string>
```

To get a user's access information for a dataset:

```
GET /datasets/<id>/acls/<userid> HTTP/1.1
Host: DOMAIN
Authorization: <authorization_string>
```

To get a user's access information for a committed datatype:

```
GET /datatypes/<id>/acls/<userid> HTTP/1.1
Host: DOMAIN
Authorization: <authorization_string>
```

where:

- <id> is the UUID of the requested dataset/group/committed datatype
- <userid> is the userid for the requested user. Use the special userid “default” to get the default access permissions for the object

Request Parameters

This implementation of the operation does not use request parameters.

Request Headers

This implementation of the operation uses only the request headers that are common to most requests. See *Common Request Headers*

Responses

Response Headers

This implementation of the operation uses only response headers that are common to most responses. See *Common Response Headers*.

Response Elements

On success, a JSON response will be returned with the following elements:

acl

A JSON object that describes a user's access permissions. Subkeys of `acl` are:

`userName`: the userid of the requested user

`create`: A boolean flag that indicates if the user is authorized to create new resources

`delete`: A boolean flag that indicates if the user is authorized to delete resources

`read`: A boolean flag that indicates if the user is authorized to read (GET) resources

`update`: A boolean flag that indicates if the user is authorized to update resources

`readACL`: A boolean flag that indicates if the user is authorized to read the object's ACL

`updateACL`: A boolean flag that indicates if the user is authorized to update the object's ACL

hrefs

An array of hypertext links to related resources. See *Hypermedia*.

Special Errors

This implementation of the operation does not return special errors. For general information on standard error codes, see *Common Error Responses*.

Examples

Sample Request

```
GET /groups/052dcbbd-9d33-11e4-86ce-3c15c2da029e/acls/test_user1 HTTP/1.1
host: tall.test.hdfgroup.org
Accept-Encoding: gzip, deflate
Accept: */*
User-Agent: python-requests/2.3.0 CPython/2.7.8 Darwin/14.0.0
```

Sample Response

```

HTTP/1.1 200 OK
Date: Fri, 16 Jan 2015 20:06:08 GMT
Content-Length: 660
Etag: "2c410d1c469786f25ed0075571a8e7a3f313cec1"
Content-Type: application/json
Server: TornadoServer/3.2.2

```

```

{
  "acl": {
    "create": false,
    "delete": false,
    "read": true,
    "readACL": false,
    "update": false,
    "updateACL": false,
    "userName": "test_user1"
  },
  "hrefs": [
    {
      "href": "http://tall_acl.test.hdfgroup.org/groups/eb8f6959-8775-11e5-96b6-
↪3c15c2da029e/acls/test_user1",
      "rel": "self"
    },
    {
      "href": "http://tall_acl.test.hdfgroup.org/groups/eb8f6959-8775-11e5-96b6-
↪3c15c2da029e",
      "rel": "root"
    },
    {
      "href": "http://tall_acl.test.hdfgroup.org/",
      "rel": "home"
    },
    {
      "href": "http://tall_acl.test.hdfgroup.org/groups/eb8f6959-8775-11e5-96b6-
↪3c15c2da029e",
      "rel": "owner"
    }
  ]
}

```

Related Resources

- *PUT ACL*
- *GET ACLs*

8.3.2 GET ACLs

Description

Returns access information for all users defined in the ACL (Access Control List) for the object with the UUID provided in the URI.

Requests

Syntax

To get the ACL for a domain:

```
GET /acls HTTP/1.1
Host: DOMAIN
Authorization: <authorization_string>
```

To get the ACL for a group:

```
GET /groups/<id>/acls HTTP/1.1
Host: DOMAIN
Authorization: <authorization_string>
```

To get the ACL for a dataset:

```
GET /datasets/<id>/acls HTTP/1.1
Host: DOMAIN
Authorization: <authorization_string>
```

To get the ACL for a committed datatype:

```
GET /datatypes/<id>/acls HTTP/1.1
Host: DOMAIN
Authorization: <authorization_string>
```

where:

- <id> is the UUID of the requested dataset/group/committed datatype

Request Parameters

This implementation of the operation does not use request parameters.

Request Headers

This implementation of the operation uses only the request headers that are common to most requests. See *Common Request Headers*

Responses

Response Headers

This implementation of the operation uses only response headers that are common to most responses. See *Common Response Headers*.

Response Elements

On success, a JSON response will be returned with the following elements:

acls

A JSON list that contains one element for each user specified in the ACL. The elements will be JSON objects that describe each user's access permissions. The subkeys of each element are:

`userName`: the userid of the user ('default' for the default access)

`create`: A boolean flag that indicates if the user is authorized to create new resources

`delete`: A boolean flag that indicates if the user is authorized to delete resources

`read`: A boolean flag that indicates if the user is authorized to read (GET) resources

`update`: A boolean flag that indicates if the user is authorized to update resources

`readACL`: A boolean flag that indicates if the user is authorized to read the object's ACL

`updateACL`: A boolean flag that indicates if the user is authorized to update the object's ACL

hrefs

An array of hypertext links to related resources. See *Hypermedia*.

Special Errors

This implementation of the operation does not return special errors. For general information on standard error codes, see *Common Error Responses*.

Examples

Sample Request

```
GET /groups/052dcbbd-9d33-11e4-86ce-3c15c2da029e/acls HTTP/1.1
host: tall.test.hdfgroup.org
Accept-Encoding: gzip, deflate
Accept: */*
User-Agent: python-requests/2.3.0 CPython/2.7.8 Darwin/14.0.0
```

Sample Response

```
HTTP/1.1 200 OK
Date: Fri, 16 Jan 2015 20:06:08 GMT
Content-Length: 660
Etag: "2c410d1c469786f25ed0075571a8e7a3f313cec1"
Content-Type: application/json
Server: TornadoServer/3.2.2
```

```
{
  "acls": [
    {
      "create": true,
      "delete": true,
```

(continues on next page)

(continued from previous page)

```
    "read": true,
    "readACL": true,
    "update": true,
    "updateACL": true,
    "userName": "test_user2"
  },
  {
    "create": false,
    "delete": false,
    "read": true,
    "readACL": false,
    "update": false,
    "updateACL": false,
    "userName": "test_user1"
  },
  {
    "create": false,
    "delete": false,
    "read": false,
    "readACL": false,
    "update": false,
    "updateACL": false,
    "userName": "default"
  }
],
"hrefs": [
  {
    "href": "http://tall_acl.test.hdfgroup.org/groups/eb8f6959-8775-11e5-96b6-
↪3c15c2da029e/acls",
    "rel": "self"
  },
  {
    "href": "http://tall_acl.test.hdfgroup.org/groups/eb8f6959-8775-11e5-96b6-
↪3c15c2da029e",
    "rel": "root"
  },
  {
    "href": "http://tall_acl.test.hdfgroup.org/",
    "rel": "home"
  },
  {
    "href": "http://tall_acl.test.hdfgroup.org/groups/eb8f6959-8775-11e5-96b6-
↪3c15c2da029e",
    "rel": "owner"
  }
]
```

Related Resources

- *PUT ACL*
- *GET ACL*

8.3.3 PUT ACL

Description

Update the access information for the given user for the object with the UUID provided in the URI.

Requests

Syntax

To update a user's access information for a domain:

```
PUT /acls/<userid> HTTP/1.1
Host: DOMAIN
Authorization: <authorization_string>
```

To update a user's access information for a group:

```
PUT /groups/<id>/acls/<userid> HTTP/1.1
Host: DOMAIN
Authorization: <authorization_string>
```

To get a user's access information for a dataset:

```
PUT /datasets/<id>/acls/<userid> HTTP/1.1
Host: DOMAIN
Authorization: <authorization_string>
```

To get a user's access information for a committed datatype:

```
PUT /datatypes/<id>/acls/<userid> HTTP/1.1
Host: DOMAIN
Authorization: <authorization_string>
```

where:

- <id> is the UUID of the requested dataset/group/committed datatype
- <userid> is the userid for the requested user. Use the special userid "default" to get the default access permissions for the object

Request Parameters

This implementation of the operation does not use request parameters.

Request Headers

This implementation of the operation uses only the request headers that are common to most requests. See *Common Request Headers*

Request Elements

The request body must include a JSON object that has the following keys and boolean values:

```
{ 'read': <True or False>,
  'create': <True or False>,
  'update': <True or False>,
  'delete': <True or False>,
  'readACL': <True or False>,
  'updateACL': <True or False>
}
```

Responses

Response Headers

This implementation of the operation uses only response headers that are common to most responses. See [Common Response Headers](#).

Response Elements

On success, a JSON response will be returned with the following elements:

hrefs

An array of hypertext links to related resources. See [Hypermedia](#).

Special Errors

This implementation of the operation does not return special errors. For general information on standard error codes, see [Common Error Responses](#).

Examples

Sample Request

```
PUT /groups/052dcbbd-9d33-11e4-86ce-3c15c2da029e/acls/test_user1 HTTP/1.1
host: tall.test.hdfgroup.org
Accept-Encoding: gzip, deflate
Accept: */*
User-Agent: python-requests/2.3.0 CPython/2.7.8 Darwin/14.0.0

{ 'read': True, 'create': False, 'update': False,
  'delete': False, 'readACL': False, 'updateACL': False }
```

Sample Response

```
HTTP/1.1 201 Created
Date: Fri, 16 Jan 2015 20:06:08 GMT
Content-Length: 660
Etag: "2c410dlc469786f25ed0075571a8e7a3f313cec1"
Content-Type: application/json
Server: TornadoServer/3.2.2
```

```
"hrefs": [
  {
    "href": "http://tall_acl.test.hdfgroup.org/groups/eb8f6959-8775-11e5-96b6-
↪3c15c2da029e/acls/test_user1",
    "rel": "self"
  },
  {
    "href": "http://tall_acl.test.hdfgroup.org/groups/eb8f6959-8775-11e5-96b6-
↪3c15c2da029e",
    "rel": "root"
  },
  {
    "href": "http://tall_acl.test.hdfgroup.org/",
    "rel": "home"
  },
  {
    "href": "http://tall_acl.test.hdfgroup.org/groups/eb8f6959-8775-11e5-96b6-
↪3c15c2da029e",
    "rel": "owner"
  }
]
```

Related Resources

- [GET ACL](#)
- [GET ACLs](#)

9.1 Authorization and Authentication

9.1.1 Request Authentication

The HDF REST API supports HTTP Basic authentication to authenticate users by comparing an encrypted username and password against a value stored within a password file.

If neither the requested object (Group, Dataset, or Committed Datatype) nor the object's root group has an Access Control List (ACL), authorization is not required and no authentication string needs to be supplied. See *Access Control List* for information on how to use ACL's.

If the requested object (or object's root group) does have an ACL, authorization may be required (if the object is not publically readable), and if so the requestor will need to provide an Authorization header in the request. If authorization is required, but not provided, the server will return an HTTP Status of 401 - Unauthorized.

If authorization is required (i.e. a 401 response is received), the client should provide an authorization header in the http request which conveys the userid and password.

The authorization string is constructed as follows:

1. Username and password are combined into a string "username:password". Note that username cannot contain the ":" character
2. The resulting string is then encoded using the RFC2045-MIME variant of Base64, except not limited to 76 char/line
3. The authorization method and a space i.e. "Basic " is then put before the encoded string

For example, if the user agent uses 'Aladdin' as the username and 'open sesame' as the password then the field is formed as follows: `Authorization: Basic QWxhZGRpbjpvYGVuIHNLc2FtZQ==`. When passwords are being sent over an open network, SSL connections should be used to avoid "man in the middle attacks". The Base64 encoding is easily reversible and if using plain http there is no assurance that the password will not be compromised.

If the authorization string is validated, the server will verify the request is authorized as per the object's ACL list. If not authorized a http status 403 - Forbidden will be returned.

9.2 Common Request Headers

The following describe common HTTP request headers as used in the HDF REST API:

- **Request line:** The first line of the request, the format is of the form HTTP verb (GET, PUT, DELETE, or POST) followed by the path to the resource (e.g. /group/<uuid>. Some operations take one or more query parameters (see relevant documentation)
- **Accept:** Specifies the media type that is acceptable for the response. Valid values are “application/json”, and “/”. In addition, GET/PUT/POST Value (see *GET Value*, *PUT Value*, *POST Value*) support the value “application/octet-stream”
- **Authorization:** A string that provides the requester’s credentials for the request. See *Authorization and Authentication*
- **Host:** the domain (i.e. related collection of groups, datasets, and attributes) that the request should apply to

Note: the host header can also be provided as a query parameter. Example: <https://data.hdfgroup.org:7258/?host=tall.test.data.hdfgroup.org>

9.3 Common Response Headers

The following describes some of the common response lines returned by the HDF REST API.

- **Status Line: the first line of the response will always be: “HTTP/1.1” followed by a status code** (e.g. 200) followed by a reason message (e.g. “OK”). For errors, an additional error message may be included on this line.
- **Content-Length:** the response size in bytes.
- **Etag: a hash code that indicates the state of the requested resource. If the client** sees the same Etag value for the same request, it can assume the resource has not changed since the last request.
- **Content-Type:** the mime type of the response.

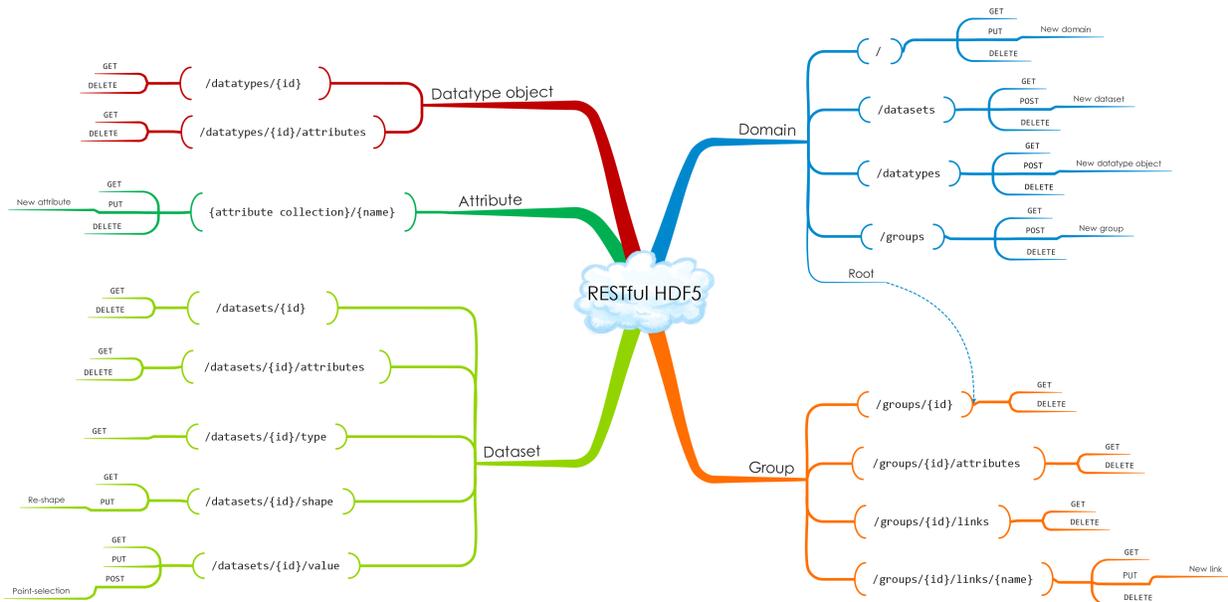
9.4 Common Error Responses

For each request, the HDF REST API returns a standard HTTP status code as described below. In general 2xx codes indicate success, 3xx codes are returned for some form of redirection, 4xx codes are returned for a client error, and 5xx codes are returned for server errors. In addition to the numeric code, the HDF REST API shall return an informational message as part of the response providing further information on the nature of the error.

- 200 OK - The request was completed successfully
- 201 Created - The request was fulfilled and a new resource (e.g. group, dataset, attribute) was created
- 400 Bad Request - The request was not structured correctly (e.g. a required key was missing).
- 401 Unauthorization - User authentication is required, supply an Authentication header with a valid username and password
- 403 Forbidden - The requesting user does not have access to the requested resource
- 404 Not Found - The requested resource was not found (e.g. GET /groups/<id> where <id> was not a valid identifier for a group in the domain).

- 405 Method Not Allowed - The HTTP request type was not valid for the given resource (e.g. PUT / datasets/<id>/type)
- 409 Conflict - This error is used with PUT requests where the resources cannot be created because there is an existing resource with the same name (e.g. PUT / where the requested domain is already present).
- 410 Gone - The resource requested has been recently deleted.
- 500 Internal Error - An unexpected error that indicates some problem occurred on the server.
- 501 Not Implemented - The request depends on a feature that is not yet implemented.

9.5 Diagram of REST operations



9.6 Hypermedia

The HDF REST API supports the REST convention of **HATEOAS** or *Hypermedia as the Engine of Application State*. The idea (see <http://en.wikipedia.org/wiki/HATEOS> for a full explanation) is that each response includes links to resources related to the requested resource.

For example, consider the request for a dataset: GET /datasets/<id>. The response will be a JSON representation of the dataset describing its type, shape, and other aspects. Related resources to the dataset would include:

- the dataset's attributes
- the dataset's value
- the dataset collection of the domain
- the root group of the domain the dataset is in
- the domain resource

So the GET /datasets/<id> response includes a key `hrefs` that contains a JSON array. Each array element has a key: `href` - the related resource, and a key: `rel` that denotes the type of relation. Example:

```

{
  "hrefs": [
    {"href": "http://tall.test.hdfgroup.org/datasets/<id>", "rel": "self"},
    {"href": "http://tall.test.hdfgroup.org/groups/<id>", "rel": "root"},
    {"href": "http://tall.test.hdfgroup.org/datasets/<id>/attributes", "rel":
↪"attributes"},
    {"href": "http://tall.test.hdfgroup.org/datasets/<id>/value", "rel": "data"},
    {"href": "http://tall.test.hdfgroup.org/", "rel": "home"}
  ]
}

```

This enables clients to “explore” the API without detailed knowledge of it.

This is the list of relations used in the HDF REST API:

- attributes - the attributes of the resource
- data - the resource’s data (used for datasets)
- database - the collection of all datasets in the domain
- groupbase - the collection of all groups in the domain
- home - the domain the resource is a member of
- owner - the containing object of this resource (e.g. the group an attribute is a member of)
- root - the root group of the domain the resource is a member of
- self - this resource
- typebase - the collection of all committed types in the domain

9.7 Resource List

9.7.1 List of Resources

Resource	GET	PUT	POST	DELETE	Description
Domain	Y	Y	N	Y	A related collection of HDF objects
Group	Y	N	N	Y	Represents an HDF Group
Links	Y	N	N	N	Collection of links within a group
Link	Y	Y	N	Y	Represents an HDF link
Dataset	Y	N	N	Y	Represents an HDF Dataset
Attributes	Y	N	N	N	Collection of Attributes
Attribute	Y	Y	N	Y	Represents an HDF Attribute
Dataspace	Y	Y	N	N	Shape of a dataset
Type	Y	N	N	N	Type of a dataset
Value	Y	Y	Y	N	Data values of a dataset
Datatype	Y	N	N	Y	Committed datatype
Groups	Y	N	Y	N	Collection of groups within a domain
Datasets	Y	N	Y	N	Collection of datasets within a domain
Datatypes	Y	N	Y	N	Collection of datatypes within a domain

9.8 Using Iteration

There are some operations that may return an arbitrarily large list of results. For example: `GET /groups/<id>/attributes` returns all the attributes of the group object with the given id. It's possible (if not common in practice) that the group may contain hundreds or more attributes.

If you desire to retrieve the list of attributes in batches (say you are developing a user interface that has a “get next page” style button), you can use iteration.

This is accomplished by adding query parameters to the request that limit the number of items returned and a marker parameter that identifies where the iteration should start off.

Let's flesh out our example by supposing the group with UUID `<id>` has 1000 attributes named “a0000”, “a0001”, and so on.

If we'd like to retrieve just the first 100 attributes, we can add a limit value to the request like so:

```
GET /groups/<id>/attributes?Limit=100
```

Now the response will return attributes “a0000”, “a0001”, through “a0099”.

To get the next hundred, use the URL-encoded name of the last attribute received as the marker value for the next request:

```
GET /groups/<id>/attributes?Limit=100&Marker="a0099"
```

This request will return attributes “a0100”, “a0101”, through “a0199”.

Repeat this pattern until less than ‘Limit’ items are returned. This indicates that you've completed the iteration through all elements of the group.

Iteration is also supported for links in a group, and the groups, datasets, and datatypes collections.

9.8.1 Related Resources

- *GET Attributes*
- *GET Groups*
- *GET Links*
- *GET Datasets*
- *GET Datatypes*

CHAPTER 10

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)