
h5serv Documentation

Release 0.1

HDFGroup

Jan 06, 2018

Contents

1	Introduction	3
2	Installation	5
2.1	Installing h5serv	5
3	Domains	11
3.1	Mapping of file paths to domain names	11
3.2	Creating Domains	12
3.3	Getting Information about Domains	12
3.4	Deleting Domains	12
3.5	List of Operations	12
4	Groups	19
4.1	Creating Groups	19
4.2	Getting information about Groups	19
4.3	Updating Links	19
4.4	Deleting Groups	20
4.5	List of Operations	20
5	Datasets	45
5.1	Creating Datasets	45
5.2	Getting information about a dataset	45
5.3	Writing data to a dataset	45
5.4	Reading data from a dataset	46
5.5	Resizable datasets	46
5.6	Deleting datasets	46
5.7	List of Operations	46
6	Committed Datatypes	81
6.1	Creating committed datatypes	81
6.2	Getting information about a committed datatype	81
6.3	Deleting committed datatypes	81
6.4	List of Operations	82
7	Attributes	93
7.1	Creating Attributes	93
7.2	Reading and Writing Data	93

7.3	Listing attributes	93
7.4	Deleting Attributes	93
7.5	List of Operations	94
8	Types	109
8.1	Predefined Types	109
8.2	String Types - Fixed Length	110
8.3	String Types - Variable Length	111
8.4	Compound Types	111
8.5	Enumerated Types	112
8.6	Array Types	113
8.7	Opaque Types	113
8.8	Object Reference Types	113
8.9	Region Reference Types	114
8.10	Type Keys	115
8.11	Related Resources	116
9	Access Control List	119
9.1	Example	119
9.2	Root ACL Inheritance	120
9.3	List of Operations	121
10	Reference	131
10.1	Authorization and Authentication	131
10.2	Common Request Headers	132
10.3	Common Response Headers	132
10.4	Common Error Responses	132
10.5	Diagram of REST operations	133
10.6	Hypermedia	133
10.7	Resource List	134
10.8	Using Iteration	134
11	Utilities	137
11.1	dumpobjdb.py	137
11.2	exportjson.py	137
11.3	exporth5.py	138
11.4	jsontoh5.py	138
11.5	h5tojson.py	138
12	Admin Tools	141
12.1	makepwd_file.py	141
12.2	update_pwd.py	141
12.3	getacl.py	142
12.4	setacl.py	142
13	What's New	145
13.1	h5serv 1.1	145
13.2	h5serv 1.0	145
14	Tutorials	147
14.1	Samples	147
15	FAQ	149
15.1	What datatypes are supported?	149
15.2	Why does h5serv use those long ids?	149

15.3	How can I get a dataset (or group) via a pathname?	150
15.4	How do I guard against an attribute (dataset/group/file) from being deleted by a request?	150
15.5	How can I display my data in a nice Web UI?	150
15.6	I have a C or Fortran application that uses HDF5, how can I take advantage of h5serv?	150
15.7	Is there documentation on the JSON format generated by h5tojson.py:	150
15.8	How do I submit a bug report?	150
16	License and Legal Info	151
16.1	Copyright Notice and License Terms for h5serv Software Service, Libraries and Utilities	151
17	Indices and tables	153

This is the developer documentation for h5serv, a Webservice for HDF5 content.

Contents:

CHAPTER 1

Introduction

h5serv is a web service that can be used to send and receive HDF5 data. h5serv uses a REST interface to support CRUD (create, read, update, delete) operations on the full spectrum of HDF5 objects including: groups, links, datasets, attributes, and committed data types. As a REST-based service a variety of clients can be developed in JavaScript, Python, C, and other common languages.

2.1 Installing h5serv

You should find h5serv quite easy to setup. The server (based on Python Tornado) is self-contained, so you will not need to setup Apache or other web server software to utilize h5serv.

2.1.1 Prerequisites

A computer running a 64-bit version of Windows, Mac OS X, or Linux.

You will also need the following Python packages:

- Python 2.7 or later
- NumPy 1.10.4 or later
- h5py 2.5 or later
- tornado 4.0.2 or later
- watchdog 0.8.3 or later
- requests 2.3 or later (for client tests)

If you are not familiar with installing Python packages, the easiest route is to use a package manager such as Anaconda (as described below).

If you have a git client installed on your system, you can directly download the h5serv source from GitHub: `git clone --recursive https://github.com/HDFGroup/h5serv.git`. Otherwise, you can download a zip file of the source from GitHub (as described below).

2.1.2 Installing on Windows

Anaconda from Continuum Analytics can be used to easily manage the package dependencies needed for HDF Server.

In a browser go to: <http://continuum.io/downloads> and click the “Windows 64-bit Python 2.7 Graphical Installer” button.

Install Anaconda using the default options.

Once Anaconda is installed select “Anaconda Command Prompt” from the start menu.

In the command window that appears, create a new anaconda environment using the following command: `conda create -n h5serv python=2.7 h5py tornado requests pytz`

Answer ‘y’ to the prompt, and the packages will be fetched.

In the same command window, run: `activate h5serv`

Install the watchdog package (this is currently not available through Anaconda): `pip install watchdog`

Download the hdf5-json project: `git clone https://github.com/HDFGroup/hdf5-json.git`. Alternatively, in a browser go to: <https://github.com/HDFGroup/hdf5-json> and click the “Download ZIP” button (right side of page). Download the zip file and extract to the destination directory of your choice.

Next, cd to the hdf5-json folder and run: `python setup.py install`.

Download the h5serv project: `git clone https://github.com/HDFGroup/h5serv.git`. Alternatively, download the source zip as described above.

Next, in the command window, cd to the folder you extracted the source files to.

From here cd to “h5serv-master/server” (or just server if you used git).

Run: `python app.py` You should see the output: “Starting event loop on port: 5000”.

You may then see a security alert: “Windows Firewall has blocked some features of this program”. Click “Allow access” with the default option (Private network access).

At this point the server is running, waiting on any requests being sent to port 5000. Go to the “verification” section below to try out the service.

2.1.3 Installing on Linux/Mac OS X

Anaconda from Continuum Analytics can be used to easily manage the package dependencies needed for HDF Server.

In a browser go to: <http://continuum.io/downloads> and click the “Mac OS X 64-bit Python 2.7 Graphical Installer” button for Mac OS X or: “Linux 64-bit Python 2.7”.

Install Anaconda using the default options.

Once Anaconda is installed, open a new shell and run the following on the command line:

```
conda create -n h5serv python=2.7 h5py tornado requests pytz
```

Answer ‘y’ to the prompt, and the packages will be fetched.

Install the watchdog package (this is currently not available through Anaconda): `pip install watchdog`

In the same shell, run: `source activate h5serv`

Download the hdf5-json project: `git clone https://github.com/HDFGroup/hdf5-json.git`. Alternatively, in a browser go to: <https://github.com/HDFGroup/hdf5-json> and click the “Download ZIP” button (right side of page). Download the zip file and extract to the destination directory of your choice.

Next, cd to the hdf5-json folder and run: `python setup.py install`.

Download the h5serv project: `git clone https://github.com/HDFGroup/h5serv.git`. Alternatively, download the source zip as described above.

Next, cd to the h5serv folder.

From here `cd` to “server” (or “h5serv-master/server” if you extracted from ZIP file).

Run: `python app.py` You should see the output: “Starting event loop on port: 5000”.

At this point the server is running, waiting on any requests being sent to port 5000. Go to the “verification” section below to try out the service.

2.1.4 Verification

To verify that h5serv was installed correctly, you can run the test suite included with the installation.

Open a new shell (on Windows, run “Anaconda Command Prompt” from the start menu).

In this shell, run the following commands:

- `source activate h5serv` (just: activate h5serv on Windows)
- `cd <h5serv installation directory>`
- `cd test`
- `python testall.py`

All tests should report OK.

2.1.5 Server Configuration

The file `h5serv/server/config.py` provides several configuration options that can be used to customize h5serv. Each of the options can be changed by:

- Changing the value in the `config.py` file and re-starting the service.
- Passing a command line option to `app.py` on startup. E.g. `python app.py --port=7253`
- Setting an environment variable with the option name in upper case. E.g. `export PORT=5000; python app.py`

The config options are:

port

The port that h5serv will listen on. Change this if 5000 conflicts with another service.

Default: 5000

debug

If `True` the server will report debug info (e.g. a stack trace) to the requester on error. If `False`, just the status code and message will be reported.

Default: `True`

datapath

A path indicating the directory where HDF5 files will be stored.

Note: Any HDF5 file content that you put in this directory will be exposed via the server REST api (unless the domain's ACL is configured to prevent public access, see: ../AclOps).

Default: ../data/

public_dir

A list of directories under datapath which will be visible to any authenticated user's request.

Default: ['public', 'test']

domain

The base DNS path for domain access (see comment to hdf5_ext config option).

Default: hdfgroup.org

hdf5_ext

The extension to assume for HDF5 files. The REST requests don't assume an extension, so a request such as:

```
GET /
HOST: tall.data.hdfgroup.org
```

Translates to: "Get the file tall.h5 in the directory given by datapath".

Default: .h5

toc_name

Name of the auto-generated HDF5 that provides a "Table Of Contents" list of all HDF5 files in the datapath directory and sub-directories.

Default: .toc.h5

home_dir

A directory under data_path that will be the parent directory of user home directories. For example if datapath is ../data, home_dir is home, the authenticated request of GET / for userid knuth would return a list of files in the directory: ../data/home/knuth.

Default: home

ssl_port

The SSL port the server will listen on for HTTPS requests.

Default: 6050

ssl_cert

Location of the SSL cert.

default:

ssl_key

The SSL key.

default:

ssl_cert_pwd

The SSL cert password

default:

password_uri

Resource path to be used for user authentication. Currently two methods are supported:

HDF5 Password file: An HDF5 that contains userids and (encrypted) passwords. See: *Admin Tools*. In this case the password_uri config is a path to the password file.

MongoDB: A MongoDB database that contains a “users” collection of userids and passwords. In this case the password_uri would be of the form: `mongodb://<mongo_ip>:<port>` where `<mongo_ip>` is the IP address of the host running the mongo database and `<port>` is the port of the mongo database (typically 27017).

default: `../util/admin/passwd.h5`

mongo_dbname

Mongo database named used for MongoDB-based authentication as described above.

default: `hdfdevtest`

static_url

URI path that will be used to map any static HTML content to be displayed by the server.

default: `/views/(.*)`

static_path

File path for files (i.e. regular HTML files) to be hosted statically.

default: `../static`

cors_domain

Domains to allow for CORS (cross-origin resource sharing). Use `*` to allow any domain, `None` to disallow.

default: `*`

log_file

File path for server log files. Set to None to have log output go to standard out.

log_level

Verbosity level for logging. One of: ERROR, WARNING, INFO, DEBUG, NOTSET.

default: INFO

background_timeout

Time interval in milliseconds to check for updates in the datapath folder (e.g. a file that is added through some external process). Set to 0 to disable background processing.

default: 1000

2.1.6 Data files

Copy any HDF5 files you would like exposed by the service to the datapath directory (h5serv/data). If you do not wish to have the files updatable by the service make the files read-only.

On the first request to the service, a Table of Contents (TOC) file will be generated which will contain links to all HDF5 files in the data folder (and sub-folders).

Note: Do not modify files once they have been placed in the datapath directory. h5serv inventories new files on first access, but won't see some changes (e.g. new group is created) made to the file outside the REST api.

**Note:* HDF5 that are newly created (copied into) the datapath directory will be "noticed" by the service and added into the TOC.

In h5serv, domains are containers for related collection of resources, similar to a file in the traditional HDF5 library. In the h5serv implementation of the HDF5 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 *Host* http header or (alternatively) the Host query parameter.

3.1 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 that data directory is '/data' and an HDF5 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 access 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

3.2 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.

Use *GET Domain* to get information about a domain, including the UUID of the domain’s root group.

3.3 Getting Information about Domains

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

3.4 Deleting Domains

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

The TOC domain cannot be deleted.

3.5 List of Operations

3.5.1 DELETE Domain

Description

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

Requests

Syntax

```
DELETE / HTTP/1.1
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*.

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

The 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
Content-Length: 0
User-Agent: python-requests/2.3.0 CPython/2.7.8 Darwin/14.0.0
host: deleteme.test.hdfgroup.org
Accept: */*
```

Sample Response

```
HTTP/1.1 200 OK
Date: Fri, 16 Jan 2015 03:47:33 GMT
Content-Length: 0
Content-Type: text/html; charset=UTF-8
Server: TornadoServer/3.2.2
```

Related Resources

- *GET Domain*
- *PUT Domain*

3.5.2 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 Host line of the http header.

Alternatively, the domain can be specified as a ‘Host’ query parameter. Example: <http://127.0.0.1:7253?host=tall.data.hdfgroup.org>.

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

Requests

Syntax

```
GET / 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*.

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).

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

The 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: 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 03:51:58 GMT
Content-Length: 508
Etag: "e45bef255ffc0530c33857b88b15f551f371de38"
Content-Type: application/json
Server: TornadoServer/3.2.2
```

```
{
  "root": "052dcbbd-9d33-11e4-86ce-3c15c2da029e",
  "created": "2015-01-16T03:51:58Z",
  "lastModified": "2015-01-16T03:51:58Z",
  "hrefs": [
    {"href": "http://tall.test.hdfgroup.org/", "rel": "self"},
    {"href": "http://tall.test.hdfgroup.org/datasets", "rel": "database"},
    {"href": "http://tall.test.hdfgroup.org/groups", "rel": "groupbase"},
    {"href": "http://tall.test.hdfgroup.org/datatypes", "rel": "typebase"},
    {"href": "http://tall.test.hdfgroup.org/groups/052dcbbd-9d33-11e4-86ce-
↪3c15c2da029e", "rel": "root"}
  ]
}
```

Related Resources

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

3.5.3 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
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*

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).

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

The 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
Content-Length: 0
User-Agent: python-requests/2.3.0 CPython/2.7.8 Darwin/14.0.0
host: newfile.test.hdfgroup.org
Accept: */*
Accept-Encoding: gzip, deflate
```

Sample Response

```
HTTP/1.1 201 Created
Date: Fri, 16 Jan 2015 04:11:52 GMT
Content-Length: 523
Content-Type: application/json
Server: TornadoServer/3.2.2
```

```
{
  "root": "cd31cfdc-9d35-11e4-aa58-3c15c2da029e",
  "created": "2015-01-16T04:11:52Z",
  "lastModified": "2015-01-16T04:11:52Z",
  "hrefs": [
    {"href": "http://newfile.test.hdfgroup.org/", "rel": "self"},
```

```
{
  {"href": "http://newfile.test.hdfgroup.org/datasets", "rel": "database"},
  {"href": "http://newfile.test.hdfgroup.org/groups", "rel": "groupbase"},
  {"href": "http://newfile.test.hdfgroup.org/datatypes", "rel": "typebase"},
  {"href": "http://newfile.test.hdfgroup.org/groups/cd31cfdc-9d35-11e4-aa58-
↪3c15c2da029e", "rel": "root"}
}
```

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.

4.1 Creating Groups

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

4.2 Getting information about Groups

Use *GET Group* to get information about a group: attribute count, link count, 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*.

4.3 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.

4.4 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 points to. These objects may become *anonymous*, i.e. they are not referenced by any link, but can still be accessed via GET request with the object uuid.

4.5 List of Operations

4.5.1 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.

Requests

Syntax

```
DELETE /groups/<id> HTTP/1.1
Host: DOMAIN
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

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

hrefs

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

Special Errors

The 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/45a882e1-9d01-11e4-8acf-3c15c2da029e HTTP/1.1
Host: testGroupDelete.test.hdfgroup.org
Authorization: authorization_string
```

Sample Response

```
HTTP/1.1 200 OK
Date: Thu, 15 Jan 2015 21:55:51 GMT
Content-Length: 270
Content-Type: application/json
Server: TornadoServer/3.2.2
```

```
{
  "hrefs": [
    {"href": "http://testGroupDelete.test.hdfgroup.org/groups", "rel": "self"},
    {"href": "http://testGroupDelete.test.hdfgroup.org/groups/45a06719-9d01-11e4-9b1c-
↪3c15c2da029e", "rel": "root"},
    {"href": "http://testGroupDelete.test.hdfgroup.org/", "rel": "home"}
  ]
}
```

Related Resources

- *POST Group*
- *GET Group*

4.5.2 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
Host: DOMAIN
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/25dd052b-a06d-11e4-a29e-3c15c2da029e/links/deleteme HTTP/1.1
Content-Length: 0
User-Agent: python-requests/2.3.0 CPython/2.7.8 Darwin/14.0.0
host: tall_updated.test.hdfgroup.org
Accept: */*
Accept-Encoding: gzip, deflate
```

Sample Response

```
HTTP/1.1 200 OK
Date: Tue, 20 Jan 2015 06:25:37 GMT
Content-Length: 299
Content-Type: application/json
Server: TornadoServer/3.2.2
```

```
{
  "hrefs": [
    {"href": "http://tall_updated.test.hdfgroup.org/groups/25dd052b-a06d-11e4-a29e-
↪3c15c2da029e", "rel": "root"},
    {"href": "http://tall_updated.test.hdfgroup.org/", "rel": "home"},
    {"href": "http://tall_updated.test.hdfgroup.org/groups/25dd052b-a06d-11e4-a29e-
↪3c15c2da029e", "rel": "owner"}
  ]
}
```

Related Resources

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

4.5.3 GET Group

Description

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

Requests

Syntax

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

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

The 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 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
```

```
{
  "id": "052dcbbd-9d33-11e4-86ce-3c15c2da029e",
  "attributeCount": 2,
  "linkCount": 2,
  "created": "2015-01-16T03:47:22Z",
  "lastModified": "2015-01-16T03:47:22Z",
  "hrefs": [
    { "href": "http://tall.test.hdfgroup.org/groups/052dcbbd-9d33-11e4-86ce-
    ↪3c15c2da029e", "rel": "self" },
    { "href": "http://tall.test.hdfgroup.org/groups/052dcbbd-9d33-11e4-86ce-
    ↪3c15c2da029e/links", "rel": "links" },
    { "href": "http://tall.test.hdfgroup.org/groups/052dcbbd-9d33-11e4-86ce-
    ↪3c15c2da029e", "rel": "root" },
```

```
{ "href": "http://tall.test.hdfgroup.org/", "rel": "home" },
  { "href": "http://tall.test.hdfgroup.org/groups/052dcbbd-9d33-11e4-86ce-
↪3c15c2da029e/attributes", "rel": "attributes" }
]
}
```

Related Resources

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

4.5.4 GET Groups

Description

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

Requests

Syntax

```
GET /groups HTTP/1.1
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 UUID's to return.

Marker

If provided, a string value indicating that only UUID's 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 (including 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

The 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: 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 21:53:48 GMT
Content-Length: 449
Etag: "83575a7865761b6d4eaf5d285ab1de062c49250b"
Content-Type: application/json
Server: TornadoServer/3.2.2
```

```
{
  "groups": [
    "052e001e-9d33-11e4-9a3d-3c15c2da029e",
    "052e13bd-9d33-11e4-91a6-3c15c2da029e",
    "052e5ae8-9d33-11e4-888d-3c15c2da029e",
    "052e700a-9d33-11e4-9fe4-3c15c2da029e",
    "052e89c7-9d33-11e4-b9bc-3c15c2da029e"
  ],
  "hrefs": [
    {"href": "http://tall.test.hdfgroup.org/groups", "rel": "self"},
    {"href": "http://tall.test.hdfgroup.org/groups/052dcbbd-9d33-11e4-86ce-
↪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 /groups?Marker=cba6e3fd-9dbd-11e4-bf4a-3c15c2da029e&Limit=5 HTTP/1.1
host: group1k.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, 16 Jan 2015 22:02:46 GMT
Content-Length: 458
Etag: "49221af3436fdaca7e26c74b491ccf8698555f08"
Content-Type: application/json
Server: TornadoServer/3.2.2
```

```
{
  "groups": [
    "cba6fc19-9dbd-11e4-846e-3c15c2da029e",
    "cba71842-9dbd-11e4-abd0-3c15c2da029e",
    "cba73442-9dbd-11e4-a6e9-3c15c2da029e",
    "cba74fc5-9dbd-11e4-bc15-3c15c2da029e",
    "cba77c2e-9dbd-11e4-9c71-3c15c2da029e"
  ],
  "hrefs": [
    {"href": "http://group1k.test.hdfgroup.org/groups", "rel": "self"},
    {"href": "http://group1k.test.hdfgroup.org/groups/cb9ebf11-9dbd-11e4-
↪9e83-3c15c2da029e", "rel": "root"},
    {"href": "http://group1k.test.hdfgroup.org/", "rel": "home"}
  ]
}
```

Related Resources

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

4.5.5 GET Link

Description

Returns information about a Link.

Requests

Syntax

```
GET /groups/<id>/links/<name> HTTP/1.1
Host: DOMAIN
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 symbol 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

The 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/052dcbbd-9d33-11e4-86ce-3c15c2da029e/links/g1 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 - Hard Link

```
HTTP/1.1 200 OK
Date: Fri, 16 Jan 2015 22:42:05 GMT
Content-Length: 688
Etag: "70c5c4f2f7cac9f7f155fe026f4c492f65e3fb8e"
Content-Type: application/json
Server: TornadoServer/3.2.2
```

```
{
  "link": {
    "title": "g1",
    "collection": "groups",
    "class": "H5L_TYPE_HARD",
    "id": "052e001e-9d33-11e4-9a3d-3c15c2da029e"
  },
  "created": "2015-01-16T03:47:22Z",
  "lastModified": "2015-01-16T03:47:22Z",
  "hrefs": [
    {"href": "http://tall.test.hdfgroup.org/groups/052dcbbd-9d33-11e4-86ce-
    ↪3c15c2da029e/links/g1", "rel": "self"},
    {"href": "http://tall.test.hdfgroup.org/groups/052dcbbd-9d33-11e4-86ce-
    ↪3c15c2da029e", "rel": "root"},
    {"href": "http://tall.test.hdfgroup.org/", "rel": "home"},
    {"href": "http://tall.test.hdfgroup.org/groups/052dcbbd-9d33-11e4-86ce-
    ↪3c15c2da029e", "rel": "owner"},
    {"href": "http://tall.test.hdfgroup.org/groups/052e001e-9d33-11e4-9a3d-
    ↪3c15c2da029e", "rel": "target"}
  ]
}
```

Sample Request - Soft Link

```
GET /groups/052e700a-9d33-11e4-9fe4-3c15c2da029e/links/slink 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
Related Resources
```

Sample Response - Soft Link

```
HTTP/1.1 200 OK
Date: Fri, 16 Jan 2015 23:29:27 GMT
Content-Length: 620
Etag: "7bd777729ac5af261c85c7e3b87ef0045739bf77"
Content-Type: application/json
Server: TornadoServer/3.2.2
```

```
{
  "link": {
    "title": "slink",
    "class": "H5L_TYPE_SOFT",
    "h5path": "somevalue"
  },
  "created": "2015-01-16T03:47:22Z",
  "lastModified": "2015-01-16T03:47:22Z",
  "hrefs": [
    {"href": "http://tall.test.hdfgroup.org/groups/052e700a-9d33-11e4-9fe4-3c15c2da029e/links/slink", "rel": "self"},
    {"href": "http://tall.test.hdfgroup.org/groups/052dcbbd-9d33-11e4-86ce-3c15c2da029e", "rel": "root"},
    {"href": "http://tall.test.hdfgroup.org/", "rel": "home"},
    {"href": "http://tall.test.hdfgroup.org/groups/052e700a-9d33-11e4-9fe4-3c15c2da029e", "rel": "owner"},
    {"href": "http://tall.test.hdfgroup.org/#h5path(somevalue)", "rel": "target"}
  ]
}
```

Sample Request - External Link

```
GET /groups/052e5ae8-9d33-11e4-888d-3c15c2da029e/links/extlink 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 - External Link

```
HTTP/1.1 200 OK
Date: Tue, 20 Jan 2015 05:47:55 GMT
Content-Length: 644
Etag: "1b7a228acdb19f7259ed8a1b3ba4bc442b405ef9"
Content-Type: application/json
Server: TornadoServer/3.2.2
```

```
{
  "link": {
    "title": "extlink",
    "class": "H5L_TYPE_EXTERNAL",
    "h5path": "somepath",
    "h5domain": "somefile"
  },
  "created": "2015-01-16T03:47:22Z",
  "lastModified": "2015-01-16T03:47:22Z",
  "hrefs": [
    {"href": "http://tall.test.hdfgroup.org/groups/052e5ae8-9d33-11e4-888d-
↪3c15c2da029e/links/extlink", "rel": "self"},
    {"href": "http://tall.test.hdfgroup.org/groups/052dcbbd-9d33-11e4-86ce-
↪3c15c2da029e", "rel": "root"},
    {"href": "http://tall.test.hdfgroup.org/", "rel": "home"},
    {"href": "http://tall.test.hdfgroup.org/groups/052e5ae8-9d33-11e4-888d-
↪3c15c2da029e", "rel": "owner"},
    {"href": "http://somefile.hdfgroup.org#h5path(somepath)", "rel": "target"}
  ]
}
```

Sample Request - User Defined Link

```
GET /groups/0262c3a6-a069-11e4-8905-3c15c2da029e/links/udlink HTTP/1.1
host: tall_with_udlink.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 - 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: TornadoServer/3.2.2
```

```
{
  "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"},
    {"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*

4.5.6 GET Links

Description

Returns all the links for a given group.

Requests

Syntax

```
GET /groups/<id>/links HTTP/1.1
Host: DOMAIN
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

The 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/0ad37be1-a06f-11e4-8651-3c15c2da029e/links 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: Tue, 20 Jan 2015 06:55:19 GMT
Content-Length: 607
Etag: "49edcce6a8f724108d41d52c98002d6255286ff8"
Content-Type: application/json
Server: TornadoServer/3.2.2
```

```
{
  "links": [
    {
      "title": "g1.2.1",
      "class": "H5L_TYPE_HARD",
      "collection": "groups",
      "id": "0ad38d45-a06f-11e4-a909-3c15c2da029e"
    },
    {
      "title": "extlink",
      "class": "H5L_TYPE_EXTERNAL",
      "h5path": "somepath",
      "file": "somefile"
    }
  ],
  "hrefs": [
    {"href": "http://tall.test.hdfgroup.org/groups/0ad37be1-a06f-11e4-8651-3c15c2da029e/links", "rel": "self"},
    {"href": "http://tall.test.hdfgroup.org/groups/0ad2e151-a06f-11e4-bc68-3c15c2da029e", "rel": "root"},
    {"href": "http://tall.test.hdfgroup.org/", "rel": "home"},
    {"href": "http://tall.test.hdfgroup.org/groups/0ad37be1-a06f-11e4-8651-3c15c2da029e", "rel": "owner"}
  ]
}
```

Sample Request Batch

```
GET /groups/76bddb1e-a06e-11e4-86d6-3c15c2da029e/links?Marker=g0089&Limit=5 HTTP/1.1
host: group1k.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 Batch

```
HTTP/1.1 200 OK
Date: Tue, 20 Jan 2015 07:30:03 GMT
Content-Length: 996
Etag: "221affdeae54076d3493ce8ce0ed80ddb89c6e27"
Content-Type: application/json
Server: TornadoServer/3.2.2
```

```
{
  "links": [
    {"title": "g0090", "id": "76c53485-a06e-11e4-96f3-3c15c2da029e", "class": "H5L_
    ↳TYPE_HARD", "collection": "groups"},
    {"title": "g0091", "id": "76c54d40-a06e-11e4-a342-3c15c2da029e", "class": "H5L_
    ↳TYPE_HARD", "collection": "groups"},
    {"title": "g0092", "id": "76c564f5-a06e-11e4-bccd-3c15c2da029e", "class": "H5L_
    ↳TYPE_HARD", "collection": "groups"},
    {"title": "g0093", "id": "76c57d19-a06e-11e4-a9a8-3c15c2da029e", "class": "H5L_
    ↳TYPE_HARD", "collection": "groups"},
  ]
}
```

```

    {"title": "g0094", "id": "76c5941c-a06e-11e4-b641-3c15c2da029e", "class": "H5L_
↪TYPE_HARD", "collection": "groups"}
  ],
  "hrefs": [
    {"href": "http://group1k.test.hdfgroup.org/groups/76bddb1e-a06e-11e4-86d6-
↪3c15c2da029e/links", "rel": "self"},
    {"href": "http://group1k.test.hdfgroup.org/groups/76bddb1e-a06e-11e4-86d6-
↪3c15c2da029e", "rel": "root"},
    {"href": "http://group1k.test.hdfgroup.org/", "rel": "home"},
    {"href": "http://group1k.test.hdfgroup.org/groups/76bddb1e-a06e-11e4-86d6-
↪3c15c2da029e", "rel": "owner"}
  ]
}

```

Related Resources

- *DELETE Link*
- *GET Link*
- *GET Group*
- *PUT Link*

4.5.7 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
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

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

id

The UUID of the group 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

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 links to related resources. See *Hypermedia*.

Special Errors

The 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
Content-Length: 0
User-Agent: python-requests/2.3.0 CPython/2.7.8 Darwin/14.0.0
host: testGroupPost.test.hdfgroup.org
Accept: */*
Accept-Encoding: gzip, deflate
```

Sample Response

```
HTTP/1.1 201 Created
Content-Length: 705
Content-Location: http://testGroupPost.test.hdfgroup.org/groups/777978c5-a078-11e4-8755-3c15c2da029e
Server: TornadoServer/3.2.2
Location: http://testGroupPost.test.hdfgroup.org/groups/777978c5-a078-11e4-8755-3c15c2da029e
Date: Tue, 20 Jan 2015 07:46:38 GMT
Content-Type: application/json
```

```
{
  "id": "777978c5-a078-11e4-8755-3c15c2da029e",
  "created": "2015-01-20T07:46:38Z",
  "lastModified": "2015-01-20T07:46:38Z",
  "attributeCount": 0,
  "linkCount": 0,
  "hrefs": [
    {"href": "http://testGroupPost.test.hdfgroup.org/groups/777978c5-a078-11e4-8755-3c15c2da029e", "rel": "self"},
    {"href": "http://testGroupPost.test.hdfgroup.org/groups/777978c5-a078-11e4-8755-3c15c2da029e/links", "rel": "links"},
    {"href": "http://testGroupPost.test.hdfgroup.org/groups/777109b3-a078-11e4-8512-3c15c2da029e", "rel": "root"},
    {"href": "http://testGroupPost.test.hdfgroup.org/", "rel": "home"},
    {"href": "http://testGroupPost.test.hdfgroup.org/groups/777978c5-a078-11e4-8755-3c15c2da029e/attributes", "rel": "attributes"}
  ]
}
```

Sample Request with Link

Create a new Group, link to root (which has uuid of “36b921f3-...” as “linked_group”.

```
POST /groups HTTP/1.1
Content-Length: 79
User-Agent: python-requests/2.3.0 CPython/2.7.8 Darwin/14.0.0
host: testGroupPostWithLink.test.hdfgroup.org
Accept: */*
Accept-Encoding: gzip, deflate
```

```
{
  "link": {
    "id": "36b921f3-a07a-11e4-88da-3c15c2da029e",
    "name": "linked_group"
  }
}
```

Sample Response with Link

```
HTTP/1.1 201 Created
Content-Length: 745
Content-Location: http://testGroupPostWithLink.test.hdfgroup.org/groups/36cbe08a-a07a-
↪11e4-8301-3c15c2da029e
Server: TornadoServer/3.2.2
Location: http://testGroupPostWithLink.test.hdfgroup.org/groups/36cbe08a-a07a-11e4-
↪8301-3c15c2da029e
Date: Tue, 20 Jan 2015 07:59:09 GMT
Content-Type: application/json
```

```
{
  "id": "36cbe08a-a07a-11e4-8301-3c15c2da029e",
  "attributeCount": 0,
  "linkCount": 0,
  "created": "2015-01-20T07:59:09Z",
  "lastModified": "2015-01-20T07:59:09Z",
  "hrefs": [
    {"href": "http://testGroupPostWithLink.test.hdfgroup.org/groups/36cbe08a-a07a-
↪11e4-8301-3c15c2da029e", "rel": "self"},
    {"href": "http://testGroupPostWithLink.test.hdfgroup.org/groups/36cbe08a-a07a-
↪11e4-8301-3c15c2da029e/links", "rel": "links"},
    {"href": "http://testGroupPostWithLink.test.hdfgroup.org/groups/36b921f3-a07a-
↪11e4-88da-3c15c2da029e", "rel": "root"},
    {"href": "http://testGroupPostWithLink.test.hdfgroup.org/", "rel": "home"},
    {"href": "http://testGroupPostWithLink.test.hdfgroup.org/groups/36cbe08a-a07a-
↪11e4-8301-3c15c2da029e/attributes", "rel": "attributes"}
  ]
}
```

Related Resources

- *DELETE Group*
- *GET Links*

- *PUT Link*
- *GET Group*
- *GET Groups*

4.5.8 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
Host: DOMAIN
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 key:

id

The UUID of the group 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. If this key is present, the `h5path` and `h5domain` keys will be ignored

h5path

A string describing a path to an external resource. If this key is present an soft or external link will be created.

h5domain

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

The 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 “e0309a0a-...”, create a hard link named “g3” that points to the object with uuid “e032ad9c-...”.

```
PUT /groups/e0309a0a-a198-11e4-b127-3c15c2da029e/links/g3 HTTP/1.1
Content-Length: 46
User-Agent: python-requests/2.3.0 CPython/2.7.8 Darwin/14.0.0
host: tall_updated.test.hdfgroup.org
Accept: */*
Accept-Encoding: gzip, deflate
```

```
{"id": "e032ad9c-a198-11e4-8d53-3c15c2da029e"}
```

Sample Response - Create Hard Link

```
HTTP/1.1 201 Created
Date: Wed, 21 Jan 2015 18:11:09 GMT
Content-Length: 418
Content-Type: application/json
Server: TornadoServer/3.2.2
```

```
{
  "hrefs": [
    { "href": "http://tall_updated.test.hdfgroup.org/groups/e0309a0a-a198-11e4-b127-
↪3c15c2da029e/links/g3", "rel": "self"},
    { "href": "http://tall_updated.test.hdfgroup.org/groups/e0309a0a-a198-11e4-b127-
↪3c15c2da029e", "rel": "root"},
    { "href": "http://tall_updated.test.hdfgroup.org/", "rel": "home"},
    { "href": "http://tall_updated.test.hdfgroup.org/groups/e0309a0a-a198-11e4-b127-
↪3c15c2da029e", "rel": "owner"}
  ]
}
```

Sample Request - Create Soft Link

In group “e0309a0a-...”, create a soft link named “softlink” that contains the path “/somewhere”.

```
PUT /groups/e0309a0a-a198-11e4-b127-3c15c2da029e/links/softlink HTTP/1.1
Content-Length: 24
User-Agent: python-requests/2.3.0 CPython/2.7.8 Darwin/14.0.0
host: tall_updated.test.hdfgroup.org
Accept: */*
Accept-Encoding: gzip, deflate
```

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

Sample Response - Create Soft Link

```
HTTP/1.1 201 Created
Date: Wed, 21 Jan 2015 18:35:26 GMT
Content-Length: 424
Content-Type: application/json
Server: TornadoServer/3.2.2
```

```
{
  "hrefs": [
    { "href": "http://tall_updated.test.hdfgroup.org/groups/e0309a0a-a198-11e4-b127-
↪3c15c2da029e/links/softlink", "rel": "self"},
    { "href": "http://tall_updated.test.hdfgroup.org/groups/e0309a0a-a198-11e4-b127-
↪3c15c2da029e", "rel": "root"},
    { "href": "http://tall_updated.test.hdfgroup.org/", "rel": "home"},
    { "href": "http://tall_updated.test.hdfgroup.org/groups/e0309a0a-a198-11e4-b127-
↪3c15c2da029e", "rel": "owner"}
  ]
}
```

Sample Request - Create External Link

In group “d2f8bd6b-...”, create an external link named “extlink” that references the object at path: “/somewhere” in domain: “external_target.test.hdfgroup.org”.

```
PUT /groups/d2f8bd6b-a1b1-11e4-ae1c-3c15c2da029e/links/extlink HTTP/1.1
Content-Length: 69
User-Agent: python-requests/2.3.0 CPython/2.7.8 Darwin/14.0.0
host: tall_updated.test.hdfgroup.org
Accept: */*
Accept-Encoding: gzip, deflate
```

```
{"h5domain": "external_target.test.hdfgroup.org", "h5path": "/dset1"}
```

Sample Response - Create External Link

```
HTTP/1.1 201 Created
Date: Wed, 21 Jan 2015 21:09:45 GMT
Content-Length: 423
Content-Type: application/json
Server: TornadoServer/3.2.2
```

```
{
  "hrefs": [
    {"href": "http://tall_updated.test.hdfgroup.org/groups/d2f8bd6b-a1b1-11e4-ae1c-
    ↪3c15c2da029e/links/extlink", "rel": "self"},
    {"href": "http://tall_updated.test.hdfgroup.org/groups/d2f8bd6b-a1b1-11e4-ae1c-
    ↪3c15c2da029e", "rel": "root"},
    {"href": "http://tall_updated.test.hdfgroup.org/", "rel": "home"},
    {"href": "http://tall_updated.test.hdfgroup.org/groups/d2f8bd6b-a1b1-11e4-ae1c-
    ↪3c15c2da029e", "rel": "owner"}
  ]
}
```

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*.

5.1 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 filter (e.g. GZIP, LZF, SZIP).

5.2 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”.

5.3 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 is desired to update a specific list of data elements, a point selection (series of element coordinates) can be passed to the *PUT Value* operation.

5.4 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 start and end indexes 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.)

5.5 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 which will allow that dimension to be extended up to the given value. To resize the dataset, use PUT_DataShape.rst operation with the desired shape value(s) for the new dimensions.

Note: dimensions can only be increased, not decreased.

5.6 Deleting datasets

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

5.7 List of Operations

5.7.1 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

The 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*

5.7.2 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

The 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*

5.7.3 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 UUID's to return.

Marker

If provided, a string value indicating that only UUID's 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 UUID's, one for each dataset in the domain.

hrefs

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

Special Errors

The 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",
  ]
}
```

```

    "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*

5.7.4 GET Shape

Description

Gets 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 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 compatible with the dataset's type, which gives the *fill* value for the dataset (the value for 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

The 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*

5.7.5 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

The 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",
```

```
        "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*

5.7.6 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 dataset 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 indexes 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

The 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, 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],
```

```

    [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"},
  ]
}

```

```
{ "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 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: "927b5ed89616896d3dce7df8bdddac058321076a"
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: "927b5ed89616896d3dce7df8bddac058321076a"
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*

5.7.7 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 that 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 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.

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

The 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
```

```
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: resizabledset.datasettest.test.hdfgroup.org
```

```
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": [
    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](#)

5.7.8 POST Value

Description

Gets values of a data 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 consist with the dataset type (e.g. an compound type).

Special Errors

The 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*

5.7.9 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

The 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/22dff8f-a83b-11e4-883d-
↪3c15c2da029e", "rel": "root" }
  ]
}
```

Related Resources

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

- *POST Value*
- *PUT Value*

5.7.10 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 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 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 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 (which 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

The 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 datasets 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 know as “named types”), are object 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.

6.1 Creating committed datatypes

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

6.2 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 object with a “collection” key of “datatypes”.

6.3 Deleting committed datatypes

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

6.4 List of Operations

6.4.1 DELETE Datatype

Description

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

Requests

Syntax

```
DELETE /datatypes/<id> HTTP/1.1
Host: DOMAIN
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

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

hrefs

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

Special Errors

The 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/93b6a335-ac44-11e4-8d71-3c15c2da029e HTTP/1.1
Content-Length: 0
User-Agent: python-requests/2.3.0 CPython/2.7.8 Darwin/14.0.0
host: namedtype_deleted.test.hdfgroup.org
Accept: */*
Accept-Encoding: gzip, deflate
```

Sample Response

```
HTTP/1.1 200 OK
Date: Wed, 04 Feb 2015 08:05:26 GMT
Content-Length: 363
Content-Type: application/json
Server: TornadoServer/3.2.2
```

```
{
  "hrefs": [
    {"href": "http://namedtype_deleted.test.hdfgroup.org/datatypes", "rel": "self"},
    {"href": "http://namedtype_deleted.test.hdfgroup.org/", "rel": "home"},
    {"href": "http://namedtype_deleted.test.hdfgroup.org/groups/93b51245-ac44-11e4-
→8a21-3c15c2da029e", "rel": "root"}
  ]
}
```

Related Resources

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

6.4.2 GET Datatype

Description

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

Requests

Syntax

```
GET /datatypes/<id> HTTP/1.1
Host: DOMAIN
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.

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 dataset was created in UTC (ISO-8601 format).

lastModified

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

hrefs

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

Special Errors

The 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: “f545543d-...”.

Sample Request

```
GET /datatypes/f545543d-a1b4-11e4-8fa4-3c15c2da029e HTTP/1.1
host: namedtype.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, 21 Jan 2015 21:36:49 GMT
Content-Length: 619
Etag: "c53bc5b2d3c3b5059b71ef92ca7d144a2df54456"
Content-Type: application/json
Server: TornadoServer/3.2.2
```

```
{
  "id": "f545543d-a1b4-11e4-8fa4-3c15c2da029e",
  "type": {
    "base": "H5T_IEEE_F32LE",
    "class": "H5T_FLOAT"
  },
  "created": "2015-01-21T21:32:01Z",
  "lastModified": "2015-01-21T21:32:01Z",
  "attributeCount": 1,
  "hrefs": [
    {"href": "http://namedtype.test.hdfgroup.org/datatypes/f545543d-a1b4-11e4-8fa4-3c15c2da029e", "rel": "self"},
    {"href": "http://namedtype.test.hdfgroup.org/groups/f545103d-a1b4-11e4-b4a1-3c15c2da029e", "rel": "root"},
    {"href": "http://namedtype.test.hdfgroup.org/datatypes/f545543d-a1b4-11e4-8fa4-3c15c2da029e/attributes", "rel": "attributes"},
  ]
}
```

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

Related Resources

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

6.4.3 GET Datatypes

Description

Gets all the committed datatypes in a domain.

Requests

Syntax

```
GET /datatypes 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 UUID's to return.

Marker

If provided, a string value indicating that only UUID's 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

The 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: namedtype.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, 21 Jan 2015 22:42:30 GMT
Content-Length: 350
Etag: "e01f56869a9a919b1496c463f3569a2a7c319f11"
Content-Type: application/json
Server: TornadoServer/3.2.2
```

```
{
  "datatypes": [
    "f54542e6-a1b4-11e4-90bf-3c15c2da029e",
    "f545543d-a1b4-11e4-8fa4-3c15c2da029e"
  ],
  "hrefs": [
    {"href": "http://namedtype.test.hdfgroup.org/datatypes", "rel": "self"},
    {"href": "http://namedtype.test.hdfgroup.org/groups/f545103d-a1b4-11e4-b4a1-
↪3c15c2da029e", "rel": "root"},
    {"href": "http://namedtype.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. Also the “Limit” request parameter is used to limit the number of UUIDs in the response to 5.

```
GET /datatypes?Marker=d779cd5e-a1e6-11e4-8fc5-3c15c2da029e&Limit=5 HTTP/1.1
host: type1k.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: Thu, 22 Jan 2015 03:32:13 GMT
Content-Length: 461
Etag: "a2e2d5a3ae63cd504d02b51d99f27b30d17b75b5"
Content-Type: application/json
Server: TornadoServer/3.2.2
```

```
{
  "datatypes": [
    "d779ddd9-a1e6-11e4-89e5-3c15c2da029e",
    "d779ef11-a1e6-11e4-8837-3c15c2da029e",
    "d77a008a-a1e6-11e4-8840-3c15c2da029e",
    "d77a121e-a1e6-11e4-b2b0-3c15c2da029e",
    "d77a2523-a1e6-11e4-aa6d-3c15c2da029e"
  ],
  "hrefs": [
    {"href": "http://type1k.test.hdfgroup.org/datatypes", "rel": "self"},
    {"href": "http://type1k.test.hdfgroup.org/groups/d7742c14-a1e6-11e4-b2a8-
↪3c15c2da029e", "rel": "root"},
    {"href": "http://type1k.test.hdfgroup.org/", "rel": "home"}
  ]
}
```

Related Resources

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

6.4.4 POST Datatype

Description

Creates a new committed datatype.

Requests

Syntax

```
POST /datatypes 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 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.

attributeCount

The number of attributes belonging to the datatype.

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 links to related resources. See *Hypermedia*.

Special Errors

The 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
Content-Length: 26
User-Agent: python-requests/2.3.0 CPython/2.7.8 Darwin/14.0.0
host: newdtype.datatypetest.test.hdfgroup.org
Accept: */*
Accept-Encoding: gzip, deflate
```

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

Sample Response

```
HTTP/1.1 201 Created
Date: Thu, 22 Jan 2015 19:06:17 GMT
Content-Length: 533
Content-Type: application/json
Server: TornadoServer/3.2.2
```

```
{
  "id": "be08d40c-a269-11e4-84db-3c15c2da029e",
  "attributeCount": 0,
  "created": "2015-01-22T19:06:17Z",
  "lastModified": "2015-01-22T19:06:17Z",
  "hrefs": [
    { "href": "http://newdtype.datatypetest.test.hdfgroup.org/datatypes/be08d40c-a269-11e4-84db-3c15c2da029e", "rel": "self" },
    { "href": "http://newdtype.datatypetest.test.hdfgroup.org/groups/be00807d-a269-11e4-8d9c-3c15c2da029e", "rel": "root" },
    { "href": "http://newdtype.datatypetest.test.hdfgroup.org/datatypes/be08d40c-a269-11e4-84db-3c15c2da029e/attributes", "rel": "attributes" }
  ]
}
```

Sample Request with Link

Create a new committed datatype and link to root as “linked_dtype”.

```
POST /datatypes HTTP/1.1
Content-Length: 106
User-Agent: python-requests/2.3.0 CPython/2.7.8 Darwin/14.0.0
host: newlinkedtype.datatypetest.test.hdfgroup.org
Accept: */*
Accept-Encoding: gzip, deflate
```

```
{
  "type": "H5T_IEEE_F64LE",
  "link": {
    "id": "76b0bbf8-a26c-11e4-8d4c-3c15c2da029e",
    "name": "linked_dtype"
  }
}
```

Sample Response with Link

```
HTTP/1.1 201 Created
Date: Thu, 22 Jan 2015 19:25:46 GMT
Content-Length: 548
Content-Type: application/json
Server: TornadoServer/3.2.2
```

```
{
  "id": "76c3c33a-a26c-11e4-998c-3c15c2da029e",
  "attributeCount": 0,
  "created": "2015-01-22T19:25:46Z",
  "lastModified": "2015-01-22T19:25:46Z",
  "hrefs": [
    { "href": "http://newlinkedtype.datatypetest.test.hdfgroup.org/datatypes/76c3c33a-
↪a26c-11e4-998c-3c15c2da029e", "rel": "self" },
    { "href": "http://newlinkedtype.datatypetest.test.hdfgroup.org/groups/76b0bbf8-
↪a26c-11e4-8d4c-3c15c2da029e", "rel": "root" },
    { "href": "http://newlinkedtype.datatypetest.test.hdfgroup.org/datatypes/76c3c33a-
↪a26c-11e4-998c-3c15c2da029e/attributes", "rel": "attributes" }
  ]
}
```

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.

7.1 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.

7.2 Reading and Writing Data

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

7.3 Listing attributes

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

7.4 Deleting Attributes

Use *DELETE Attribute* to delete an attribute.

7.5 List of Operations

7.5.1 DELETE Attribute

Description

The implementation of the DELETE operation deletes the attribute named in the URI. All attributes and links of the dataset will also be deleted.

Requests

Syntax

```
DELETE /groups/<id>/<name> HTTP/1.1
Host: DOMAIN
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

The 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/36ae688a-ac0e-11e4-a44b-3c15c2da029e/attributes/attr1 HTTP/1.1
Content-Length: 0
User-Agent: python-requests/2.3.0 CPython/2.7.8 Darwin/14.0.0
host: tall_updated.test.hdfgroup.org
Accept: */*
Accept-Encoding: gzip, deflate
```

Sample Response

```
HTTP/1.1 200 OK
Date: Wed, 04 Feb 2015 01:36:17 GMT
Content-Length: 420
Content-Type: application/json
Server: TornadoServer/3.2.2
```

```
{
  "hrefs": [
    {"href": "http://tall_updated.test.hdfgroup.org/groups/36ae688a-ac0e-11e4-a44b-
↪3c15c2da029e/attributes", "rel": "self"},
    {"href": "http://tall_updated.test.hdfgroup.org/groups/36ae688a-ac0e-11e4-a44b-
↪3c15c2da029e", "rel": "owner"},
    {"href": "http://tall_updated.test.hdfgroup.org/groups/36ae688a-ac0e-11e4-a44b-
↪3c15c2da029e", "rel": "root"},
    {"href": "http://tall_updated.test.hdfgroup.org/", "rel": "home"}
  ]
}
```

Related Resources

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

7.5.2 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
Host: DOMAIN
Authorization: <authorization_string>
```

To get an attribute of a dataset:

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

To get an attribute of a datatype:

```
GET /datatypes/<id>/attributes/<name> HTTP/1.1
Host: DOMAIN
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 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

The 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: “45a882e1-...”.

```
GET /groups/1a956e54-abf6-11e4-b878-3c15c2da029e/attributes/attr1 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: Tue, 03 Feb 2015 22:44:04 GMT
Content-Length: 648
Etag: "55b2e2ce2d3a2449a49cfd76c4dae635ec43a150"
Content-Type: application/json
Server: TornadoServer/3.2.2
```

```
{
  "name": "attr1",
  "type": {
    "class": "H5T_INTEGER",
    "base": "H5T_STD_I8LE"
  },
  "shape": {
    "class": "H5S_SIMPLE",
    "dims": [10]
  },
  "value": [97, 98, 99, 100, 101, 102, 103, 104, 105, 0],
  "created": "2015-02-03T22:40:09Z",
  "lastModified": "2015-02-03T22:40:09Z",
  "hrefs": [
    {"href": "http://tall.test.hdfgroup.org/groups/1a956e54-abf6-11e4-b878-3c15c2da029e/attributes/attr1", "rel": "self"},
    {"href": "http://tall.test.hdfgroup.org/groups/1a956e54-abf6-11e4-b878-3c15c2da029e", "rel": "owner"},
    {"href": "http://tall.test.hdfgroup.org/groups/1a956e54-abf6-11e4-b878-3c15c2da029e", "rel": "root"},
    {"href": "http://tall.test.hdfgroup.org/", "rel": "home"}
  ]
}
```

Related Resources

- [DELETE Attribute](#)
- [GET Attributes](#)
- [GET Dataset](#)
- [GET Datatype](#)
- [GET Group](#)
- [PUT Attribute](#)

7.5.3 GET Attributes

Description

Gets all the attributes of a dataset, group, or committed datatype. For each attribute the request returns the attributes 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
Host: DOMAIN
Authorization: <authorization_string>
```

To get the attributes of a dataset:

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

To get the attributes of a datatype:

```
GET /datatypes/<id>/attributes HTTP/1.1
Host: DOMAIN
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

The 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: “45a882e1-...”.

```
GET /groups/1a956e54-abf6-11e4-b878-3c15c2da029e/attributes 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, 04 Feb 2015 00:49:28 GMT
Content-Length: 807
Etag: "7cbeefcf8d9997a8865bdea3bf2d541a14e9bf71"
Content-Type: application/json
Server: TornadoServer/3.2.2
```

```
{
  "attributes": [
    {
      "name": "attr1",
      "type": {
        "base": "H5T_STD_I8LE",
        "class": "H5T_INTEGER"
      },
      "shape": {
        "dims": [10],
```

```

    "class": "H5S_SIMPLE"
  },
  "created": "2015-02-03T22:40:09Z",
  "lastModified": "2015-02-03T22:40:09Z",
},
"name": "attr2",
"type": {
  "base": "H5T_STD_I32BE",
  "class": "H5T_INTEGER"
},
"shape": {
  "dims": [2, 2],
  "class": "H5S_SIMPLE"
},
"created": "2015-02-03T22:40:09Z",
"lastModified": "2015-02-03T22:40:09Z",
}
],
"hrefs": [
  {"href": "http://tall.test.hdfgroup.org/groups/1a956e54-abf6-11e4-b878-
↪3c15c2da029e/attributes", "rel": "self"},
  {"href": "http://tall.test.hdfgroup.org/groups/1a956e54-abf6-11e4-b878-
↪3c15c2da029e", "rel": "owner"},
  {"href": "http://tall.test.hdfgroup.org/groups/1a956e54-abf6-11e4-b878-
↪3c15c2da029e", "rel": "root"},
  {"href": "http://tall.test.hdfgroup.org/", "rel": "home"}
]
}

```

Sample Request - get Batch

Get 5 the five attributes that occur after attribute “a0004” from a of a group with UUID: “45a882e1-...”.

```

GET /groups/4cecd4dc-ac0a-11e4-af59-3c15c2da029e/attributes?Marker=a0004&Limit=5 HTTP/
↪1.1
host: attr1k.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 - get Batch

```

HTTP/1.1 200 OK
Date: Wed, 04 Feb 2015 01:08:16 GMT
Content-Length: 1767
Etag: "9483f4356e08d12b719aa64ece09e659b05adaf2"
Content-Type: application/json
Server: TornadoServer/3.2.2

```

```

{
  "attributes": [
    {
      "name": "a0005",
      "type": {"cset": "H5T_CSET_ASCII", "order": "H5T_ORDER_NONE", "class": "H5T_STRING
↪", "strpad": "H5T_STR_NULLTERM", "strsize": "H5T_VARIABLE"},

```

```

    "shape": {"class": "H5S_SCALAR"},
    "created": "2015-02-03T22:40:09Z",
    "lastModified": "2015-02-03T22:40:09Z"
  }, {
    "name": "a0006",
    "type": {"cset": "H5T_CSET_ASCII", "order": "H5T_ORDER_NONE", "class": "H5T_STRING"},
    ↪ "strpad": "H5T_STR_NULLTERM", "strsize": "H5T_VARIABLE"},
    "shape": {"class": "H5S_SCALAR"},
    "created": "2015-02-03T22:40:09Z",
    "lastModified": "2015-02-03T22:40:09Z"
  }, {
    "name": "a0007",
    "type": {"cset": "H5T_CSET_ASCII", "order": "H5T_ORDER_NONE", "class": "H5T_STRING"},
    ↪ "strpad": "H5T_STR_NULLTERM", "strsize": "H5T_VARIABLE"},
    "shape": {"class": "H5S_SCALAR"},
    "created": "2015-02-03T22:40:09Z",
    "lastModified": "2015-02-03T22:40:09Z"
  }, {
    "name": "a0008",
    "type": {"cset": "H5T_CSET_ASCII", "order": "H5T_ORDER_NONE", "class": "H5T_STRING"},
    ↪ "strpad": "H5T_STR_NULLTERM", "strsize": "H5T_VARIABLE"},
    "shape": {"class": "H5S_SCALAR"},
    "created": "2015-02-03T22:40:09Z",
    "lastModified": "2015-02-03T22:40:09Z"
  }, {
    "name": "a0009",
    "type": {"cset": "H5T_CSET_ASCII", "order": "H5T_ORDER_NONE", "class": "H5T_STRING"},
    ↪ "strpad": "H5T_STR_NULLTERM", "strsize": "H5T_VARIABLE"},
    "shape": {"class": "H5S_SCALAR"},
    "created": "2015-02-03T22:40:09Z",
    "lastModified": "2015-02-03T22:40:09Z"
  }
],
"hrefs": [
  {"href": "http://attr1k.test.hdfgroup.org/groups/4cecd4dc-ac0a-11e4-af59-3c15c2da029e/attributes", "rel": "self"},
  ↪ {"href": "http://attr1k.test.hdfgroup.org/groups/4cecd4dc-ac0a-11e4-af59-3c15c2da029e", "rel": "owner"},
  ↪ {"href": "http://attr1k.test.hdfgroup.org/groups/4cecd4dc-ac0a-11e4-af59-3c15c2da029e", "rel": "root"},
  {"href": "http://attr1k.test.hdfgroup.org/", "rel": "home"}
]
}

```

Related Resources

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

7.5.4 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
Host: DOMAIN
Authorization: <authorization_string>
```

To create a dataset attribute:

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

To create a committed datatype attribute:

```
PUT /datatypes/<id>/attributes/<name> HTTP/1.1
Host: DOMAIN
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 provide to make a non-scalar attribute.

type

Specify’s 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

The 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 “be319519-” named “attr4”. The value of the attribute will be 42.

```
PUT /groups/be319519-acff-11e4-bf8e-3c15c2da029e/attributes/attr4 HTTP/1.1
Content-Length: 38
User-Agent: python-requests/2.3.0 CPython/2.7.8 Darwin/14.0.0
host: tall_updated.test.hdfgroup.org
Accept: */*
Accept-Encoding: gzip, deflate
```

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

Sample Response - scalar attribute

```
HTTP/1.1 201 Created
Date: Thu, 05 Feb 2015 06:25:30 GMT
Content-Length: 359
Content-Type: application/json
Server: TornadoServer/3.2.2
```

```
{ "hrefs": [
  { "href": "http://tall_updated.test.hdfgroup.org/groups/be319519-acff-11e4-bf8e-
  ↪3c15c2da029e/attributes/attr4", "rel": "self"},
  { "href": "http://tall_updated.test.hdfgroup.org/groups/be319519-acff-11e4-bf8e-
  ↪3c15c2da029e", "rel": "owner"},
  { "href": "http://tall_updated.test.hdfgroup.org/groups/be319519-acff-11e4-bf8e-
  ↪3c15c2da029e", "rel": "root"}
  ]
}
```

Sample Request - string attribute

Create a two-element, fixed width string attribute in the group with UUID of “be319519-” named “attr6”. The attributes values will be “Hello, ...” and “Goodbye!”.

```
PUT /groups/be319519-acff-11e4-bf8e-3c15c2da029e/attributes/attr6 HTTP/1.1
Content-Length: 162
User-Agent: python-requests/2.3.0 CPython/2.7.8 Darwin/14.0.0
host: tall_updated.test.hdfgroup.org
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 Response - string attribute

```
HTTP/1.1 201 Created
Date: Thu, 05 Feb 2015 06:42:14 GMT
Content-Length: 359
```

```
Content-Type: application/json
Server: TornadoServer/3.2.2
```

```
{
  "hrefs": [
    { "href": "http://tall_updated.test.hdfgroup.org/groups/be319519-acff-11e4-bf8e-
    ↪3c15c2da029e/attributes/attr6", "rel": "self" },
    { "href": "http://tall_updated.test.hdfgroup.org/groups/be319519-acff-11e4-bf8e-
    ↪3c15c2da029e", "rel": "owner" },
    { "href": "http://tall_updated.test.hdfgroup.org/groups/be319519-acff-11e4-bf8e-
    ↪3c15c2da029e", "rel": "root" }
  ]
}
```

Sample Request - compound type

Create a two-element, attribute of group with UUID of “be319519-” named “attr_compound”. The attribute has a compound type with an integer and a floating point element.

```
PUT /groups/be319519-acff-11e4-bf8e-3c15c2da029e/attributes/attr_compound HTTP/1.1
Content-Length: 187
User-Agent: python-requests/2.3.0 CPython/2.7.8 Darwin/14.0.0
host: tall_updated.test.hdfgroup.org
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 Response - compound type

```
HTTP/1.1 201 Created
Date: Thu, 05 Feb 2015 06:49:19 GMT
Content-Length: 367
Content-Type: application/json
Server: TornadoServer/3.2.2
```

```
{
  "hrefs": [
    { "href": "http://tall_updated.test.hdfgroup.org/groups/be319519-acff-11e4-bf8e-
    ↪3c15c2da029e/attributes/attr_compound", "rel": "self" },
    { "href": "http://tall_updated.test.hdfgroup.org/groups/be319519-acff-11e4-bf8e-
    ↪3c15c2da029e", "rel": "owner" },
    { "href": "http://tall_updated.test.hdfgroup.org/groups/be319519-acff-11e4-bf8e-
    ↪3c15c2da029e", "rel": "root" }
  ]
}
```

```
]
}
```

Related Resources

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

The h5serv 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).

8.1 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_U6 {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}`: a eight byte unsigned integer
- `H5T_STD_I64 {LE | BE}`: a eight byte signed integer
- `H5T_IEEE_F32 {LE | BE}`: a four byte floating-point value
- `H5T_IEEE_F64 {LE | BE}`: a eight byte floating-point integer

Predefined types ending in “LE” or 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 are used to specify the byte ordering in the server storage system and are not reflected in the JSON representation of the values.

8.1.1 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]
}
```

8.2 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"

8.2.1 Example

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

```
{
  "id": "1e8a359c-ac46-11e4-9f3e-3c15c2da029e",
  "shape": {
    "class": "H5S_SCALAR",
  },
  "type": {
    "class": "H5T_STRING",
    "charSet": "H5T_CSET_ASCII",
    "strPad": "H5T_STR_NULLPAD",
    "length": 40
  },
  "value": "Hello, World!"
}
```

8.3 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 know 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”

8.3.1 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."
  ]
}
```

8.4 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 an dataset or attribute’s values.

8.4.1 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 ]
}

```

8.5 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 list strings with their associated numeric values.

8.5.1 Example

```

{
  "id": "1e8a359c-ac46-11e4-9f3e-3c15c2da029e",
  "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,
      "SOLID": 0
    }
  }
}

```

```

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

```

8.6 Array Types

Array types are used when it is desired for each element of a 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.

8.6.1 Example

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

```

{
  "id": "9348ad51-7bf7-11e4-a66f-3c15c2da029e",
  "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] ]
  ]
}

```

8.7 Opaque Types

TBD

8.7.1 Example

TBD

8.8 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 a 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.

8.8.1 Example

A JSON representation of an attribute that consist 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/a09a9b99-7bf7-11e4-aa4b-3c15c2da029e",
    "",
    "datasets/a09a8efa-7bf7-11e4-9fb6-3c15c2da029e"
  ]
}
```

8.9 Region Reference Types

A region reference types 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).

8.9.1 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 “68ee967a-...”.

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": "68ee8647-7bed-11e4-9397-3c15c2da029e",
  "shape": {
    "class": "H5S_SIMPLE",
    "dims": [2]
  },
  "type": {
```

```

    "class": "H5T_REFERENCE",
    "base": "H5T_STD_REF_DSETREG"
  },
  "value": [
    {
      "id": "68ee967a-7bed-11e4-819c-3c15c2da029e",
      "select_type": "H5S_SEL_POINTS",
      "selection": [
        [0, 1], [2, 11], [1, 0], [2, 4]
      ]
    },
    {
      "id": "68ee967a-7bed-11e4-819c-3c15c2da029e",
      "select_type": "H5S_SEL_HYPERSLABS",
      "selection": [
        [ [0, 0], [0, 2] ],
        [ [0, 11], [0, 13] ],
        [ [2, 0], [2, 2] ],
        [ [2, 11], [2, 13] ]
      ]
    }
  ]
}

```

8.10 Type Keys

Information on the JSON keys used in type specifications.

8.10.1 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

8.10.2 base

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

8.10.3 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

8.10.4 charSet

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

8.10.5 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

8.10.6 length

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

8.10.7 name

The field name for compound types.

8.10.8 mapping

The enum name for enum types.

8.10.9 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.

8.11 Related Resources

- *GET Dataset*
- *GET Type*
- *POST Dataset*
- *GET Attribute*

- *PUT Attribute*
- *GET Datatype*
- *POST Datatype*

Access Control List

Access Control List (ACL) are key-value stores 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, 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. ../DatasetOpsPUT_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 or not list in the ACL (including un-authenticated requests that don't provide a username).

9.1 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 your are", while HTTP Status 403 says: "I know who you are, but you don't have permissions to access this resource."

9.2 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 of 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

9.3 List of Operations

9.3.1 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 describe a users acces permissions. Subkeys of acl are:

userName: the userid of the requested user

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

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

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

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

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

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

hrefs

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

Special Errors

The 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*

9.3.2 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 object that describe the users access permissions. Subkeys of the element are are:

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

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

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

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

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

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

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

hrefs

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

Special Errors

The 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,
      "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*

9.3.3 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

The 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](#)

10.1 Authorization and Authentication

10.1.1 Request Authentication

h5serv supports HTTP Basic authentication to authenticate users by comparing an encrypted username and password against a value stored within a password file. (See *Admin Tools* to create a password file and add user accounts.)

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 `./AclOps`) 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 QWxhZGRpbjpvYGVuIHhlc2FtZQ==`. 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.

10.1.2 User ids and passwords

User ids and passwords are maintained in an HDF5 file referenced in the server config: 'password_file'. The admin tool (See *Admin Tools*) script: update_pwd.py can be used to create new users and update passwords.

10.2 Common Request Headers

The following describe common HTTP request headers as used in h5serv:

- **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:** Specified the media type that is acceptable for the response. Valid values are "application/json", and "/. In addition, GET Value (see *GET Value*) supports 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 paramter. Example: <https://data.hdfgroup.org:7258/?host=tall.test.data.hdfgroup.org>

10.3 Common Response Headers

The following describes some of the common response lines returned by h5serv.

- **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 changes since the last request.
- **Content-Type:** the mime type of the response. Currently always "application/json".

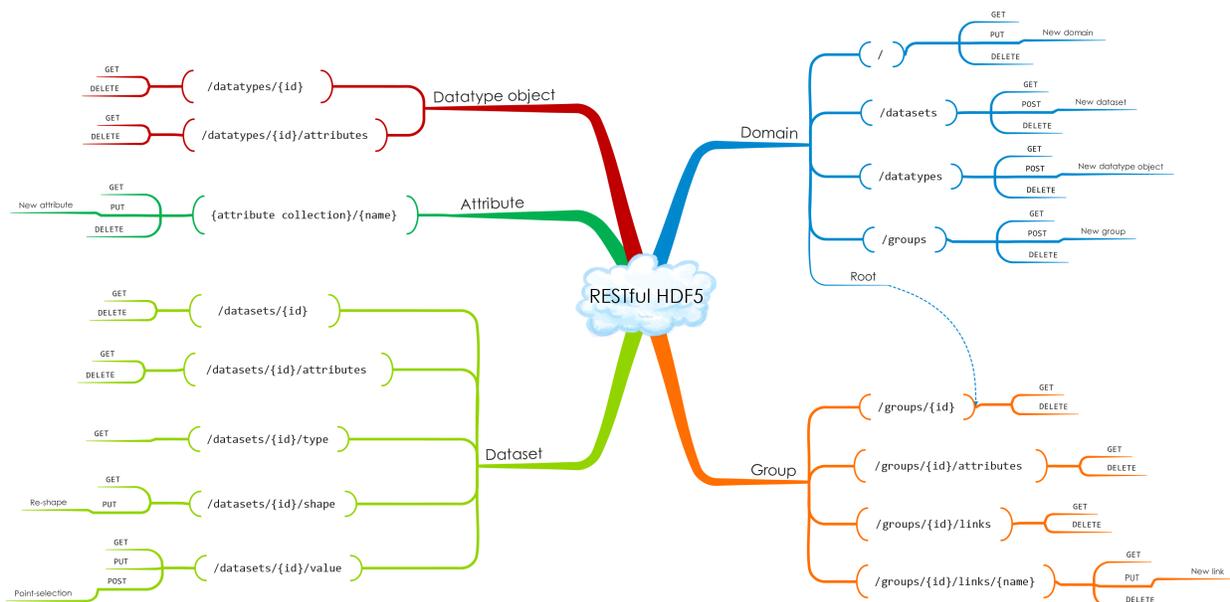
10.4 Common Error Responses

For each request, h5serv returns a standard HTTP status code as described below. In general 2xx codes indicate success, 3xx codes some form of redirection, 4xx codes client error, and 5xx codes for server errors. In addition to the numeric code, h5serv will 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 Unauthorized - Use authentication is required, supply an Authentication header with valid user 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).
- 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.

10.5 Diagram of REST operations



10.6 Hypermedia

h5serv supports the REST convention of **HATEOAS** or *Hypermedia as the Engine of Application State*. The idea is (see <http://en.wikipedia.org/wiki/HATEOS> for a full explanation) is that each response include links to related 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 it's 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 an 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 the API.

This is the list of relations used in h5serv:

- attributes - the attributes of the resource
- data - the resources 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

10.7 Resource List

10.7.1 List of Resources

10.8 Using Iteration

There are some operations that may return an arbitrary 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 the limit the number of items returned and a marker parameter that identifies where the iteration should start off.

Let’s flush 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 the 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.

10.8.1 Related Resources

- *GET Attributes*
- *GET Groups*
- *GET Links*
- *GET Datasets*
- *GET Datatypes*

The h5serv distribution includes the following utility scripts. These are all located in the `util` directory.

11.1 `dumpobjdb.py`

This script prints all the UUID's in an h5serv data file.

Usage:

```
python dumpobjdb.py <hdf5_file>
```

`hdf5_file` is a file from the h5serv data directory. Output is a list of All UUID's and a path to the associated object.

11.2 `exportjson.py`

This script makes a series of rest requests to the desired h5serv endpoint and constructs a JSON file representing the domain's contents.

Usage:

```
python exportjson.py [-v] [-D|d] [-endpoint=<server_ip>] [-port=<port>]
<domain>
```

Options:

- `-v`: verbose, print request and response codes from server
- `-D`: suppress all data output
- `-d`: suppress data output for datasets (but not attributes)
- `-endpoint`: specify IP endpoint of server
- `-port`: port address of server [default 7253]

Example - get 'tall' collection from HDF Group server: `python exportjson.py tall.
data.hdfgroup.org`

Example - get 'tall' collection from a local server instance (assuming the server is using port 5000): `python exportjson.py -endpoint=127.0.0.1 -port=5000 tall.
test.hdfgroup.org`

11.3 exporth5.py

This script makes a series of rest requests to the desired h5serv endpoint and constructs a HDF5 file representing the domain's contents.

usage: `python exporth5.py [-v] [-endpoint=<server_ip>] [-port=<port>] <domain>
<filename>`

Options:

- `-v`: verbose, print request and response codes from server
- `-endpoint`: specify IP endpoint of server
- `-port`: port address of server [default 7253]

Example - get 'tall' collection from HDF Group server, save to tall.h5: `python exporth5.
py tall.data.hdfgroup.org tall.h5`

Example - get 'tall' collection from a local server instance (assuming the server is using port 5000): `python exporth5.py -endpoint=127.0.0.1 -port=5000 tall.
test.hdfgroup.org tall.h5`

The following two utilities are located in hdf5-json submodule: hdf5-json/util.

11.4 jsontoh5.py

Converts a JSON representation of an HDF5 file to an HDF5 file.

Usage:

`jsontoh5.py [-h] <json_file> <h5_file>`

<json_file> is the input .json file. <h5_file> is the output file (will be created by the script)

Options:

- `-h`: prints help message

11.5 h5tojson.py

This script converts the given HDF5 file to a JSON representation of the file.

Usage:

`python h5tojson.py [-h] -[D|-d] <hdf5_file>`

Output is a file the hdf5 file base name and the extension .json.

Options:

- -h: prints help message
- -D: suppress all data output
- -d: suppress data output for datasets (but not attributes)

The scripts described here are intended to be run on the server by “privileged” users. These are all located in the `util\admin` directory.

12.1 `makepwd_file.py`

This script creates an initial password file “`passwd.h5`”. The password file will be used to manage http basic authentication. After creation, move the file into the location referenced by the ‘`password_file`’ configuration value.

Usage:

```
python makepwd_file.py
```

Use the `update_pwd.py` utility to create user accounts.

12.2 `update_pwd.py`

This script can be used to add users and passwords to the password file, list information about one or more users, or to update a user’s information (e.g. change the password).

Usage:

```
python update_pwd.py [-h] [-r] [-a] [-f FILE] [-u USER] [-p PASSWD]
```

Options:

- `-h`: print usage information
- `-r`: update a user’s entry
- `-a`: add a user (requires `-u` and `-p` options)
- `-f`: password file to be used
- `-u`: print/update information for specified user (otherwise show all users)

- `-p`: password to be set for the given users

Example - list all users `python update_pwd.py -f passwd.h5`

Example - list user ‘bob’: `python update_pwd.py -f passwd.h5 -u bob`

Example - add a user ‘ann’: `python update_pwd.py -f passwd.h5 -a -u ann -p mysecret`

Example - changes password for user ‘paul’: `python update_pwd.py -f passwd.h5 -r -u paul -p mysecret2`

Note, there is no way to display the passwords for any user. If a password is lost, that users password must be reset.

12.3 getacl.py

This script displays ACL’s of a given file or object within a file.

usage: `python getacl.py [-h] [-file <file>] [-path <h5path>] [userid_1, userid_2, ... userid_n]`

Options:

- `-h`: print usage information
- `-file`: (required) data file to be used
- `-path`: h5path to object. If not present, ACLs of the root group will be displayed
- `<userids>`: list of user ids to fetch ACLs for. If not present, ACLs for all users will be printed

Example - get all ACLs of tall.h5 root group `python getacl.py -file ../../data/tall.h5`

Example - get ACLs for userid 123 of root group in tall.h5 `python getacl.py -file ../../data/tall.h5 123`

Example - get ACLs for userid 123 of the dataset identified by path ‘g1/g1.1/dset1.1.1’
`python getacl.py -file ../../data/tall.h5 -path /g1/g1.1/dset1.1.1`

12.4 setacl.py

This script creates or modifies ACL’s of a given file or object within a file.

usage: `python setacl.py [-h] [-file <file>] [-path <h5path>] [+][crudep] [userid_1, userid_2, ... userid_n]`

Options:

- `-h`: print usage information
- `-file`: (required) data file to be used
- `-path`: h5path to object. If not present, ACLs of the root group will be modified
- `[+][crudep]`: add (+) or remove (-) permissions for Create (c), Read (r), Update (u), Delete (d), rEadAcl (e), and Putacl (p)
- `<userids>`: list of user ids to sets ACLs for. If not present, ACLs for the default user will be set.

Example - set default permission of tall.h5 to read only `python setacl.py -file .././data/tall.h5 +r-cudep`

Example - give userid 123 full control of tall.h5: `python setacl.py -file ../../data/tall.h5 +crudep 123`

Example - give userid read/update access to dataset at path '/g1/g1.1/dset1.1.1' `python setacl.py -file ../../data/tall.h5 -path /g1/g1.1/dset1.1.1 +ru-cdep 123`

13.1 h5serv 1.1

Significant features:

- Support was added for http over SSL (https)
- Support was added for authentication and simple user management
- Access Control List (ACL) - Enables access to HDF objects to be controlled for specific users

13.2 h5serv 1.0

This is the first release of h5serv.

Significant features:

- An implementation of the REST API as outlined in the RESTful HDF5 paper: http://www.hdfgroup.org/pubs/papers/RESTful_HDF5.pdf
- A simple DNS Server that maps DNS domains to HDF5 collections (see: <https://github.com/HDFGroup/dynamic-dns>)
- Utilities to convert native HDF5 files to HDF5-JSON and HDF5-JSON to HDF5
- UUID and timestamp extensions for HDF5 datasets, groups, and committed data types

14.1 Samples

TBD: Some walkthroughs here.

15.1 What datatypes are supported?

Type	Precisions
Integer	1, 2, 4 or 8 byte, BE/LE, signed/unsigned
Float	4, 8 byte, BE/LE
Compound	Arbitrary names and offsets
Strings (fixed-length)	Any length
Strings (variable-length)	Any length, ASCII
Opaque	Any length
Array	Any supported type
Enumeration	Any integer type
References	Region and object

Unsupported types:

Type	Status
HDF5 VLEN (non-string)	Coming soon!
HDF5 “time” type	
Opaque	
Bitfields	

15.2 Why does h5serv use those long ids?

h5serv uses the UUID standard (http://en.wikipedia.org/wiki/Universally_unique_identifier) to identify objects (datasets, groups, and committed datatypes) uniquely. The benefit of using UUIDs is that objects can be uniquely identified without requiring any central coordination.

15.3 How can I get a dataset (or group) via a pathname?

You will need to iterate through the path to get the UUID of each subgroup. E.g. suppose the path of interest is “/g1/g1.1” in the domain: “tall.data.hdfgroup.org”. Perform these actions to get the UUID of the group at /g1/g1.1.

1. GET // returns the UUID of the root group
2. GET /groups/<root_uuid>/links/g1 // returns the UUID of the group at “/g1”
3. GET /groups/<g1_uuid>/links/g1.1 // returns the UUID of the group at “/g1/g1.1”

15.4 How do I guard against an attribute (dataset/group/file) from being deleted by a request?

Future releases of h5serv will support authorization and permissions to protect content that shouldn't be altered.

For now the choices are:

1. Don't expose the h5serv endpoint on a non-trusted network
2. Make the files readonly
3. Make periodic backups of all data files
4. Don't share the domain name with non-trusted sources. Since h5serv doesn't provide an operation to list all domains on the server, creating a non-trivial domain name (e.g. “mydata_18494”) will be relatively secure.

15.5 How can I display my data in a nice Web UI?

There are many Javascript libraries (e.g. <http://d3js.org>) that can take the data returned by h5serv to create compelling graphics.

15.6 I have a C or Fortran application that uses HDF5, how can I take advantage of h5serv?

We are planning on creating a hdf5 library plugin that will transparently invoke the REST api. For now, you can use C-libraries such as libcurl to invoke h5serv requests.

15.7 Is there documentation on the JSON format generated by h5tojson.py:

Yes. There is a grammar published here: <http://hdf5-json.readthedocs.org/en/latest/index.html>.

15.8 How do I submit a bug report?

If you have a Github account, create an issue here: <https://github.com/HDFGroup/h5serv/issues>.

Alternatively, you send a email to the HDF Group help desk: help@hdfgroup.org.

16.1 Copyright Notice and License Terms for h5serv Software Service, Libraries and Utilities

h5serv (HDF5 REST Server) Service, Libraries and Utilities

Copyright (c) 2018, The HDF Group

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted for any purpose (including commercial purposes) provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions, and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions, and the following disclaimer in the documentation and/or materials provided with the distribution.
3. In addition, redistributions of modified forms of the source or binary code must carry prominent notices stating that the original code was changed and the date of the change.
4. All publications or advertising materials mentioning features or use of this software are asked, but not required, to acknowledge that it was developed by The HDF Group and credit the contributors.
5. Neither the name of The HDF Group, nor the name of any Contributor may be used to endorse or promote products derived from this software without specific prior written permission from The HDF Group or the Contributor, respectively.

DISCLAIMER: THIS SOFTWARE IS PROVIDED BY THE HDF GROUP AND THE CONTRIBUTORS “AS IS” WITH NO WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED. In no event shall The HDF Group or the Contributors be liable for any damages suffered by the users arising out of the use of this software, even if advised of the possibility of such damage.

CHAPTER 17

Indices and tables

- `genindex`
- `modindex`
- `search`