
RESTful HDF5

Interface Specification - Version 0.1

Gerd Heber, The HDF Group <gheber@hdfgroup.org>

Abstract

In this document, we specify a REST [Fielding2000] interface for HDF5 data stores. We describe HDF5 resources, URIs, and resource representations, and show a simple example of how to use this interface to populate an HDF5 store.

I would like to thank Mike Folk for his insightful comments and steady encouragement of this effort. I would like to thank my fellow members of Team HDF Group, who made this work possible in the first place. All remaining errors and inaccuracies are, of course, my own.

Introduction

But let your communication be, GET, PUT, POST, DELETE: For whatsoever is more than these, commeth of euill.

—Matthew 5:37, William Tyndale 1526, KJV 1611

The topic of a RESTful interface for HDF5 can be approached from many different starting points and directions. One perspective, which some HDF5 users may relate to, stems from the idea of accessing HDF5 files remotely over a network. This idea, perhaps as old as HDF5 itself, has been implemented successfully in efforts such as [OPeNDAP], [iRODS], [Pomegranate], [DIAL], and [SDB]. If we had to single out one trend to put renewed and increased emphasis on accessing HDF5 "stores" over a network, then it would be the growing proliferation of NoSQL and cloud-based solutions. It challenges the traditional notion of the HDF5 stack as a happy marriage between a file format and library. The transplantation of a self-contained, natively formatted file from a POSIX-compliant file system into an environment that favors contiguous I/O on large blocks and penalizes or lacks small-scale random I/O is a daunting task. For the kinds and quantities of data that are traditionally stored in HDF5 files, the attempt to maintain the fiction of a file system-like interface in an Internet-worked architecture is an expensive proposition of limited scalability.

The purpose of this document is to define a new HDF5 interface based on an architectural style for network-based architectures called REpresentational State Transfer or REST. [Fielding2000] Some of the projects and products mentioned earlier follow REST principles already. What makes this discussion different is that we are taking an *HDF5-centric* (as opposed to application domain-centric) view. Our goal is to propose a standard HDF5/REST interface that exposes all important characteristics of HDF5 "stores" without restrictions.

This document is not an introduction to the REST architectural style. There is no shortage of excellent material on REST (e.g., [Fielding2000], [REStCookbook], [.NET REST]). We also assume that the reader has a good grasp of the HTTP protocol. [HTTPHandbook]

Strictly speaking, there is no dependency between REST and HTTP. Nevertheless, to keep the following discussion somewhat practical and specific, we will focus on an HTTP-based REST interface for HDF5.

"The Hypertext Transfer Protocol (HTTP) has a special role in the Web architecture as both the primary application-level protocol for communication between Web components and the only protocol designed specifically for the transfer of resource representations."

—Section 6.3 [Fielding2000]

To define an (HTTP-based) REST interface for HDF5 we need to define three things:

1. HDF5 resources and the activities for accessing them
2. HDF5 resource identifiers (URIs)
3. HDF5 resource representations

Aside from supplementary material in appendices, this is very much the outline of this document.

Resources

"A resource is a conceptual mapping to a set of entities, not the entity that corresponds to the mapping at any particular point in time.

More precisely, a resource R is a temporally varying membership function $M_R(t)$, which for time t maps to a set of entities, or values, which are equivalent."

—Section 5.2.1.1 [Fielding2000]

Candidates of HDF5 resources are fairly easy to find. One would expect the "usual suspects" such as HDF5 groups, datasets, attributes, etc. A less obvious set of additional candidates emerges when contemplating the semantics of the four main HTTP methods used to exchange and manipulate representations of resources maintained on a server. The semantics of HTTP methods is constrained by safety and idempotency requirements as shown in Table 1, "Safety and idempotency of HTTP request methods". A method is *safe* iff it does not have side effects. Think of safe methods as read-only methods. A method is *idempotent* iff multiple invocations have the same effect as a single invocation. (A variable assignment or a projection are good examples of such methods.)

Table 1. Safety and idempotency of HTTP request methods

Method	Safe?	Idempotent?	Typical Use
GET	Yes	Yes	Obtain a resource representation
PUT	No	Yes	Update a value
DELETE	No	Yes	Delete a resource or empty a resource collection
POST	No	No	Create a new resource

The remainder of this section is an inventory of HDF5 resources, the request methods that the resources accept, and the media types supported for encoding representations.

Table 2. HDF5 Domain Resources

Resource	Methods	Description
HDF5 domain	GET	This resource represents an HDF5 domain.
		Media types: <code>application/json</code>
HDF5 root	GET	This resource represents the HDF5 domain root and contains a reference to the HDF5 root group.
		Media types: <code>application/json</code>
HDF5 group collection	GET, POST, DELETE	This resource represents the collection of all HDF5 groups in an HDF5 domain. Use DELETE to delete ALL groups (except the root group) in the domain. Use POST to create a new unlinked group in the domain.
		Media types: <code>application/json</code>

HDF5 group	GET, DELETE	This resource represents an HDF5 group. Use DELETE to delete the group. You cannot delete the root group.
		Media types: <code>application/json</code>
HDF5 group's participant collection	GET, DELETE	This resource represents the collection of HDF5 group participants. Use DELETE to delete ALL participants.
		Media types: <code>application/json</code>
HDF5 group participant	GET, DELETE, PUT	This resource represents a participant of an HDF5 group participant collection. Use PUT to create a new participant or update a participant's reference.
		Media types: <code>application/json</code>
HDF5 group's attribute collection	GET, DELETE	This resource represents the collection of HDF5 attributes of an HDF5 group. Use DELETE to delete ALL attributes of a group.
		Media types: <code>application/json</code>
HDF5 dataset collection	GET, DELETE, POST	This resource represents the collection of HDF5 datasets in an HDF5 domain. Use DELETE to delete ALL datasets in a domain. Use POST to create a new unlinked dataset.
		Media types: <code>application/json</code>
HDF5 dataset	GET, DELETE, PUT, POST	This resource represents an HDF5 dataset. Use PUT to update its value. (For extendible datasets this includes changing their extent.) Use POST for making point selections.
		Media types: <code>application/json</code> , <code>image/gif</code> , <code>image/jpeg</code> , <code>image/png</code>
HDF5 dataset's attribute collection	GET, DELETE	This resource represents the collection of HDF5 attributes of an HDF5 dataset. Use DELETE to delete ALL attributes of a dataset.
		Media types: <code>application/json</code>
HDF5 datatype collection	GET, DELETE, POST	This resource represents the collection of committed HDF5 datatypes in the HDF5 domain. Use DELETE to delete ALL unreferenced committed datatypes in the domain. Use POST to create a new unlinked committed datatype in the domain.
		Media types: <code>application/json</code>
HDF5 datatype	GET, DELETE	This resource represents a committed HDF5 datatype. (Only unreferenced committed HDF5 datatypes can be deleted.)
		Media types: <code>application/json</code>
HDF5 datatype's attribute collection	GET, DELETE	This resource represents the collection of HDF5 attributes of a committed HDF5 datatype. Use DELETE to delete ALL attributes of a datatype.

		Media types: <code>application/json</code>
HDF5 attribute	GET, DELETE, PUT	This resource represents an HDF5 attribute. Use PUT to create a new attribute or to update an attribute's value.
		Media types: <code>application/json</code>

Controllers

TBD (e.g., copy, move)

URIs

"REST uses a *resource identifier* to identify the particular resource involved in an interaction between components. ... The naming authority that assigned the resource identifier, making it possible to reference the resource, is responsible for maintaining the semantic validity of the mapping over time (i.e., ensuring that the membership function does not change)."

—Section 5.2.1.1 [Fielding2000]

The familiar HDF5 path names seem to be natural candidates for constructing Uniform Resource Identifiers (URI). However, this would make the URI space non-uniform and unpredictable, and couple clients and servers unnecessarily. A fair amount of *a priori* knowledge about an HDF5 domain would be required to navigate it. It should be easy for clients to discover the structure of the HDF5 path name space; they can then provide user-friendly navigation aids based on HDF5 path names as needed. However, in the absence of any predictability and stability in the URI structure they'd be hard to maintain for arbitrary HDF5 domains.

There are other reasons against exposing HDF5 path names as parts of URIs. HDF5 link names can be (almost) arbitrary strings which might lead to excessive URL (de-)encoding and defeat usability. *URIs should be designed to last a long time.* [RESTRCookbook] Changing an HDF5 path name associated with a resource does not change the resource itself. Why change the URI?

Notation

Let a RESTful HDF5 service be hosted at

```
http://HOST:PORT/PATH
```

which we'll abbreviate as DOMAIN. For example, DOMAIN could be `http://hdf5.cloudapp.net:8080/my-hdf5-domain`.

Many HDF5 entities (datasets, groups, etc.) are identified by universally unique identifiers (UUIDs). Let `{id}` denote such a UUID, e.g.,

```
aab20368-6e9c-4b91-899d-a42c9bcce117
```

Think of UUIDs as "addresses" in a large (128-bit), generic address space.

The only *named* entities in HDF5 files are attributes and links (= participants). Let `{name}` denote the **URL-encoded** form of such a name, e.g., the URL-encoded form of '*No weird stuff!*' is

No%20weird%20stuff!

Table 3. HDF5 Domain URIs

Resource	URI
	Description
HDF5 domain	DOMAIN
	Use this URI to get an HDF5 domain digest, which includes most of its metadata, but no dataset values.
	DOMAIN?id={id}
	Use this URI template to search the HDF5 domain by item ID.
	DOMAIN?view=[...]
Use this URI template to customize the HDF5 domain representation.	
HDF5 root	DOMAIN/root
	Use this URI to get a representation of the HDF5 root group.
HDF5 group collection	DOMAIN/groups
	Use this URI to get representations of the HDF5 groups in an HDF5 domain. Delete ALL (except the root group) HDF5 groups in the domain using DELETE. Create a new, unlinked HDF5 group using POST.
HDF5 group	DOMAIN/groups/{id}
	Use this URI to get a representation of an HDF5 group.
HDF5 group's participant collection	DOMAIN/groups/{id}/participants
	Use this URI to get representations of an HDF5 group's participants. Delete ALL participants using DELETE.
HDF5 group participant	DOMAIN/groups/{id}/participants/{name}
	Create a new participant or change the reference of an existing one using PUT.
HDF5 group's attribute collection	DOMAIN/groups/{id}/attributes
	Use this URI to get an HDF5 group's attribute collection. Delete ALL attributes using DELETE.
HDF5 dataset collection	DOMAIN/datasets
	Use this URI to get representations of the HDF5 datasets in this HDF5 domain. Create a new, unlinked HDF5 dataset using POST, or delete an existing HDF5 dataset using DELETE.
HDF5 dataset	DOMAIN/datasets/{id}?start=[...]&stride=[...]&count=[...]&block=[...]
	Use this URI to get a representation of an HDF5 dataset. Update its value or change its extent using PUT and make point selections using POST. Pass a simple hyperslab selection as a query.
	DOMAIN/datasets/{id}?view=noValue
	Use this URI to get a representation of an HDF5 dataset that does not include the dataset value.

HDF5 dataset's attribute collection	DOMAIN/datasets/{id}/attributes
	Use this URI to get a representation of an HDF5 dataset's attribute collection. Delete ALL attributes using DELETE.
HDF5 datatype collection	DOMAIN/datatypes
	Use this URI to get a representation of the committed HDF5 datatypes in an HDF5 domain. Create a new committed HDF5 datatype using POST, or delete ALL unreferenced HDF5 datatypes using DELETE.
HDF5 datatype	DOMAIN/datatype/{id}
	Use this URI to get information about a committed HDF5 datatype.
HDF5 datatypes's attribute collection	DOMAIN/datatypes/{id}/attributes
	Use this URI to get a representation of a committed HDF5 datatype's attribute collection. Delete ALL attributes using DELETE.
HDF5 attribute	{attribute collection}/{name}
	Use this URI to get an HDF5 attribute's representation. Update its value using PUT.

We use the abbreviation `{attribute collection}` as a URI shorthand for HDF5 attribute collections of HDF5 datasets, datatypes, and groups, i.e., it can be any of the following:

```
DOMAIN/datasets/{id}/attributes
DOMAIN/datatypes/{id}/attributes
DOMAIN/groups/{id}/attributes
```

Representations

"REST components perform actions on a resource by using a representation to capture the current or intended state of that resource and transferring that representation between components. A *representation* is a sequence of bytes, plus *representation metadata* to describe those bytes. Other commonly used but less precise names for a representation include: document, file, and HTTP message entity, instance or variant."

—Section 5.2.1.2 [Fielding2000]

JSON (`application/json`) and XML (`application/xml`) are probably the most common representation formats. For HDF5, JSON is the more natural choice and all our examples use JSON representations. A client must communicate its preferences via an HTTP `Accept` header, e.g.,

```
Accept: application/json;q=1.0, application/xml;q=0.1, */*;q=0.0
```

The server will reply with an HTTP `Content-Type` header indicating the MIME type of the representation. If no preference is expressed by the client, the default (JSON) is used. If the server does not support any of the requested formats, it replies with a `406 Not Acceptable` status code and a link to documentation describing the supported representations, e.g.,

```
406 Not Acceptable
Content-Type: application/json
Link: <DOMAIN/errors/mediatypes.html;>rel="help"
{
```

```
"message": "This server does not support XYZ. See help for alternatives."
}
```

The remainder of this section is a collection of request/response representation examples. Typically, a response consists of a representation of the resource and a collection of links and link templates to related resources. The latter stem from one of the core principles of linked data or the REST *HATEOAS* (hypermedia as the engine of application state) principle. No dead-end responses!

See Appendix A, *RESTful HDF5 Overview*, for an overview of the HDF5/REST interface. In Appendix C, *HDF5/JSON*, the different tokens used in the representations are defined.

HDF5 Domain

GET

```
# Request
GET / HTTP/1.1
Host: DOMAIN

#Response
HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": <id> ,

  "created":      <utctime> ,
  "lastModified": <utctime> ,

  "root": <idref> ,

  "groupCount":  <positive_integer> ,
  "datasetCount": <non_negative_integer> ,
  "datatypeCount": <non_negative_integer> ,

  "links": [
    { "rel": "self",      "href": "DOMAIN" } ,
    { "rel": "database", "href": "DOMAIN/datasets" } ,
    { "rel": "linkbase", "href": "DOMAIN/groups" } ,
    { "rel": "typebase", "href": "DOMAIN/datatypes" } ,
    { "rel": "root",     "href": "DOMAIN/root" }
  ]
}
```

HDF5 Root

GET

```
# Request
GET /root HTTP/1.1
Host: DOMAIN

#Response
HTTP/1.1 200 OK
```

```
Content-Type: application/json
```

```
{
  "id": <id> ,
  "created": <utctime>,
  "lastModified": <utctime>,
  "attributeCount": <non_negative_integer>,
  "participantCount": <non_negative_integer>,
  "links": [
    { "rel": "attributes", "href": "DOMAIN/groups/<id>/attributes" } ,
    { "rel": "group", "href": "DOMAIN/groups/<id>" } ,
    { "rel": "participants", "href": "DOMAIN/groups/<id>/participants" } ,
    { "rel": "root", "href": "DOMAIN/root" } ,
    { "rel": "self", "href": "DOMAIN/root" }
  ]
}
```

HDF5 Group Collection

GET

```
# Request
GET /groups HTTP/1.1
Host: DOMAIN

#Response
HTTP/1.1 200 OK
Content-Type: application/json

{
  "groups": [ <group_collection> ] ,
  "links": [
    { "rel": "self", "href": "DOMAIN/groups" } ,
    { "rel": "root", "href": "DOMAIN/root" }
  ]
}
```

POST

Use a POST request to create a new unlinked group resource.

```
# Request
POST /groups HTTP/1.1
Host: DOMAIN

#Response
HTTP/1.1 201 Created
Location: DOMAIN/groups/<id>
Content-Location: DOMAIN/groups/<id>
Content-Type: application/json

{
```



```

"id": <id> ,

"links": [
  { "rel": "attributes", "href": "DOMAIN/groups/<id>/attributes" },
  { "rel": "participants", "href": "DOMAIN/groups/<id>/participants" },
  { "rel": "root", "href": "DOMAIN/root" },
  { "rel": "self", "href": "DOMAIN/groups/<id>" }
],

"link-templates": [
  {
    "rel": "participant",
    "href": "DOMAIN/groups/{id}/participants/{name}",
    "method": "PUT",
    "title": "Link to a group"
  }
]
}

```

The link template can be used to link the new group to an existing group.

DELETE



Warning

This request results in the deletion of **ALL** groups in a domain's group collection, except the root group.

```

# Request
DELETE /groups HTTP/1.1
Host: DOMAIN

# Response
HTTP/1.1 200 OK

```

HDF5 Group

GET

```

# Request
GET /group/<id> HTTP/1.1
Host: DOMAIN

# Response
HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": <id>,

  "created": <utctime>,
  "lastModified": <utctime>,

  "attributeCount": <non_negative_integer>,
  "participantCount": <non_negative_integer>,

```

```

"links": [
  { "rel": "attributes", "href": "DOMAIN/groups/<id>/attributes" },
  { "rel": "participants", "href": "DOMAIN/groups/<id>/participants" },
  { "rel": "root", "href": "DOMAIN/root" },
  { "rel": "self", "href": "DOMAIN/groups/<id>" }
]
}

```

DELETE

Use this request to delete a group. As a side effect, all non-symbolic participations of this group in other groups will be deleted.

```

# First request
DELETE /group/<id> HTTP/1.1
Host: DOMAIN

# Response
HTTP/1.1 200 OK

# Second request
DELETE /group/<id> HTTP/1.1
Host: DOMAIN

# Response
HTTP/1.1 404 Not Found

```

HDF5 Group Participant Collection

GET

```

# Request
GET /group/<id>/participants HTTP/1.1
Host: DOMAIN

# Response
HTTP/1.1 200 OK
Content-Type: application/json

{
  "participants": [ <participant_collection> ],

  "links": [
    { "rel": "group", "href": "DOMAIN/groups/<id>" },
    { "rel": "root", "href": "DOMAIN/root" },
    { "rel": "self", "href": "DOMAIN/groups/<id>/participants" }
  ]
}

```

DELETE

```

# Request
DELETE /groups/<id>/participants HTTP/1.1
Host: DOMAIN

```

```
# Response
HTTP/1.1 200 OK
```

HDF5 Group Participant

GET

```
# Request
GET /group/<id>/participants/<name> HTTP/1.1
Host: DOMAIN

# Response
HTTP/1.1 200 OK
Content-Type: application/json

{
  "title": <name>,
  "idref": <id> | "hdf5": <h5path> | "href": <url> ,

  "links": [
    { "rel": "group",      "href": "DOMAIN/groups/<id>" },
    { "rel": "participants", "href": "DOMAIN/groups/<id>/participants" },
    { "rel": "root",      "href": "DOMAIN/root" },
    { "rel": "self",      "href": "DOMAIN/groups/<id>/participants/<name>" }
  ]
}
```

PUT

Use this request to create a new group participant. The destination or the referent can be specified as a UUID, an HDF5 path name, or a URL. This corresponds to hard, soft, and external links, respectively.

```
# Request
PUT /group/<id>/participants/<name> HTTP/1.1
Host: DOMAIN
Content-Type: application/json

{
  "idref": <id> | "hdf5": <h5path> | "href": <url>
}

# Response
HTTP/1.1 201 Created
Location: DOMAIN/groups/<id>/participants/<name>
Content-Location: DOMAIN/groups/<id>/participants/<name>
Content-Type: application/json

{
  "title": <name> ,
  "idref": <id> | "hdf5": <h5path> | "href": <url> ,

  "links": [
    { "rel": "group",      "href": "DOMAIN/groups/<id>" },
    { "rel": "participants", "href": "DOMAIN/groups/<id>/participants" },
    { "rel": "root",      "href": "DOMAIN/root" },
  ]
}
```

```

    { "rel": "self",          "href": "DOMAIN/groups/<id>/participants/<name>" }
  ]
}

```

DELETE

Use this request to delete a group participant.



Note

Deleting a group participant does **not** delete the referent.

```

# First request
DELETE /group/<id>/participants/<name> HTTP/1.1
Host: DOMAIN

# Response
HTTP/1.1 200 OK

# Second request
DELETE /group/<id>/participants/<name> HTTP/1.1
Host: DOMAIN

# Response
HTTP/1.1 404 Not Found

```

HDF5 Dataset Collection

GET

```

# Request
GET /datasets HTTP/1.1
Host: DOMAIN

#Response
HTTP/1.1 200 OK
Content-Type: application/json

{
  "datasets": [ <dataset_collection> ] ,
  "links": [
    { "rel": "root", "href": "DOMAIN/root" } ,
    { "rel": "self", "href": "DOMAIN/datasets" }
  ]
}

```

The dataset representations included in the dataset collection representation do not contain representations of the dataset values, or only reduced representations, such as the first ten elements. For the full value representation see the section called “HDF5 Dataset”.

POST

Use this call to create a new unlinked dataset.

```

# Request
POST /datasets HTTP/1.1
Host: DOMAIN
Content-Type: application/json

{
  "type": <datatype> ,
  "shape": <dataspace> ,
  "value": <value>
}

#Response
HTTP/1.1 201 Created
Location: DOMAIN/datasets/<id>
Content-Location: DOMAIN/datasets/<id>
Content-Type: application/json

{
  "id": <id> ,
  "links": [
    { "rel": "attributes", "href": "DOMAIN/datasets/<id>/attributes" } ,
    { "rel": "self",      "href": "DOMAIN/datasets/<id>" } ,
    { "rel": "root",     "href": "DOMAIN/root" }
  ],
  "link-templates": [
    {
      "rel": "participant",
      "href": "DOMAIN/groups/{id}/participants/{name}",
      "method": "PUT",
      "title": "Link to a group"
    }
  ]
}

```

We do not return a full-blown representation of the dataset, just the ID.

The response might return a **202 Accepted** status code for long running create requests.

DELETE



Warning

This request results in the deletion of **ALL** datasets in a domain's dataset collection. As a side-effect, all non-symbolic participations of datasets in groups will be deleted.

The response might return a **202 Accepted** status code for long running delete requests.

```

# Request
DELETE /datasets HTTP/1.1
Host: DOMAIN

# Response
HTTP/1.1 200 OK

```

HDF5 Dataset

GET

```
# Request
GET /datasets/<id> HTTP/1.1
Host: DOMAIN

# Response
HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": <id> ,
  "created": <utctime> ,
  "lastModified": <utctime> ,
  "attributeCount": <non_negative_integer> ,
  "type": <datatype> ,
  "shape": <dataspace> ,
  "value": <value> ,
  "links": [
    { "rel": "attributes", "href": "DOMAIN/datasets/<id>/attributes" } ,
    { "rel": "self", "href": "DOMAIN/datasets/<id>" } ,
    { "rel": "root", "href": "DOMAIN/root" }
  ]
}
```

To retrieve a simple hyperslab selection, submit a GET request with a query:

```
GET /datasets/<uuid>?start=[...]&stride=[...]&count=[...]&block=[...] HTTP/1.1
```

With an `Accept` header, a client may communicate a media type preference for the representation of the dataset value. Below is an example of requesting the dataset value of an HDF5 image [<http://www.hdfgroup.org/HDF5/doc/ADGuide/ImageSpec.html>] as a JPEG image.

```
# Request
GET /datasets/<id> HTTP/1.1
Accept: image/jpeg
Host: DOMAIN

# Response
HTTP/1.1 200 OK
Content-Type: multipart/related; boundary="json-et-image"

--json-et-image
Content-Type: application/json
{
  "id": <id> ,
  "created": <utctime> ,
```

```

"lastModified": <utctime> ,

"attributeCount": <non_negative_integer> ,

"type": <datatype> ,

"shape": <dataspace> ,

"links": [
  { "rel": "attributes", "href": "DOMAIN/datasets/<id>/attributes" } ,
  { "rel": "self",      "href": "DOMAIN/datasets/<id>" } ,
  { "rel": "root",     "href": "DOMAIN/root" }
]
}
--json-et-image
Content-Type: image/jpeg

... image here ...

--json-et-image

```

POST

To retrieve selected elements of a dataset's value (including set-theoretical combinations of hyperslabs), submit a POST request with a selection as its body.

```

# Request
POST /datasets/<uuid> HTTP/1.1
Host: DOMAIN
Content-Type: application/json

{
  "selection": { <selection> }
}

# Response
HTTP/1.1 200 OK
Content-Type: application/json

[...]
```

PUT

```

# Request
PUT /datasets/<id> HTTP/1.1
Host: DOMAIN
Content-Type: application/json

{
  "shape": { <dataspace> },
  "selection": { <selection> },
  "value": [...]
}

# Response
HTTP/1.1 204 No Content
Location: DOMAIN/datasets/<id>

```

```
Content-Location: DOMAIN/datasets/<id>
```

Extendible datasets can be resized by submitting a dataspace representation.

To update a simple hyperslab selection use PUT with a URI query:

```
PUT /datasets/<uuid>?start=[...]&stride=[...]&count=[...]&block=[...] HTTP/1.1
```

DELETE

Use this request to delete a dataset and all its non-symbolic group participations.

```
# First request
DELETE /datasets/<id> HTTP/1.1
Host: DOMAIN

# Response
HTTP/1.1 200 OK

# Second request
DELETE /datasets/<id> HTTP/1.1
Host: DOMAIN

# Response
HTTP/1.1 404 Not Found
```

HDF5 Attribute Collection

GET

```
# Request
GET /(datasets|datatypes|groups)/<id>/attributes HTTP/1.1
Host: DOMAIN

#Response
HTTP/1.1 200 OK
Content-Type: application/json

{
  "attributes": [ <attribute_collection> ] ,

  "links": [
    { "rel": "self", "href": "DOMAIN/(datasets|datatypes|groups)/<id>/attributes" },
    { "rel": "owner", "href": "DOMAIN/(datasets|datatypes|groups)/<id>" },
    { "rel": "root", "href": "DOMAIN/root" }
  ]
}
```

DELETE



Warning

This request results in the deletion of **ALL** attributes in an collection.

The response might return a 202 Accepted status code for long running deletion requests.

```
# Request
DELETE /(datasets|datatypes|groups)/<id>/attributes HTTP/1.1
Host: DOMAIN

# Response
HTTP/1.1 200 OK
```

HDF5 Attribute

GET

```
# Request
GET /(datasets|datatypes|groups)/<id>/attributes/<name> HTTP/1.1
Host: DOMAIN

# Response
HTTP/1.1 200 OK
Content-Type: application/json

{
  "name": <string> ,
  "type": <datatype> ,
  "shape": <datatspace> ,
  "value": <value> ,
  "links": [
    {"rel": "owner", "href": "DOMAIN/(datasets|datatypes|groups)/<id>" } ,
    {"rel": "root", "href": "DOMAIN/root" } ,
    {"rel": "self",
     "href": "DOMAIN/(datasets|datatypes|groups)/<id>/attributes/<string>"}
  ]
}
```

PUT

```
# Request
PUT /(datasets|datatypes|groups)/<id>/attributes/<name> HTTP/1.1
Host: DOMAIN
Content-Type: application/json

{
  "type": <datatype> ,
  "shape": <datatspace> ,
  "value": <value>
}
```

```
# Response
HTTP/1.1 201 Created
Content-Type: application/json
Location: DOMAIN/(datasets|datatypes|groups)/<id>/attributes/<name>
Content-Location: DOMAIN/(datasets|datatypes|groups)/<id>/attributes/<name>
Content-Type: application/json

{
  "links": [
    {"rel": "owner", "href": "DOMAIN/(datasets|datatypes|groups)/<id>" } ,
    {"rel": "root", "href": "DOMAIN/root" } ,
    {"rel": "self",
     "href": "DOMAIN/(datasets|datatypes|groups)/<id>/attributes/<name>"}
  ]
}
```

DELETE

```
# First request
DELETE /(datasets|datatypes|groups)/<id>/attributes/<name> HTTP/1.1
Host: DOMAIN

# Response
HTTP/1.1 200 OK

# Second request
DELETE /(datasets|datatypes|groups)/<id>/attributes/<name> HTTP/1.1
Host: DOMAIN

# Response
HTTP/1.1 404 Not Found
```

HDF5 Datatype Collection

A domain's datatype collection contains committed HDF5 datatype resources. Such datatypes can participate in groups (be linked) and have attributes.

GET

```
# Request
GET /datatypes HTTP/1.1
Host: DOMAIN

#Response
HTTP/1.1 200 OK
Content-Type: application/json

{
  "datatypes" : [ <datatype_collection> ] ,
  "links": [
    { "rel": "root", "href": "DOMAIN/root" },
    { "rel": "self", "href": "DOMAIN/datatypes" }
  ]
}
```

}

POST

```

# Request
POST /datatypes HTTP/1.1
Host: DOMAIN
Content-Type: application/json

{ "type": <datatype> }

#Response
HTTP/1.1 201 Created
Location: DOMAIN/datatypes/<id>
Content-Location: DOMAIN/datatypes/<id>
Content-Type: application/json

{
  "id": <id> ,
  "links": [
    { "rel": "attributes", "href": "DOMAIN/datatypes/<id>/attributes" } ,
    { "rel": "self", "href": "DOMAIN/datatypes/<id>" } ,
    { "rel": "root", "href": "DOMAIN/root" }
  ],
  "link-templates": [
    {
      "rel": "participant",
      "href": "DOMAIN/groups/{id}/participants/{name}",
      "method": "PUT",
      "title": "Link to a group"
    }
  ]
}

```

DELETE



Warning

This request results in the deletion of **ALL** committed datatypes in a domain's datatype collection. As a side-effect, all non-symbolic participations of datatypes in groups will be deleted.

The response might return a **202 Accepted** status code for long running deletion requests.



Note

The request fails, if one or more committed datatypes are in use by datasets or attributes in the domain.

```

# Request
DELETE /datatypes HTTP/1.1
Host: DOMAIN

```

```
# Response
HTTP/1.1 200 OK
```

HDF5 Datatype

GET

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

# Response
HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": <id> ,

  "created": <utctime> ,
  "lastModified": <utctime> ,

  "attributeCount": <non_negative_integer> ,

  "type": { <datatype> } ,

  "links": [
    { "rel": "attribute",
      "href": "DOMAIN/(datasets|datatypes|groups)/<id>/attributes/<name> } ,
    { "rel": "dataset",   "href": "DOMAIN/datasets/<id>/type } ,
    { "rel": "self",     "href": "DOMAIN/datatypes/<id>" } ,
    { "rel": "root",    "href": "DOMAIN/root } ,
    { "rel": "typebase", "href": "DOMAIN/datatypes }
  ]
}
```

DELETE

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

# Response
HTTP/1.1 200 OK

# Second request
DELETE /datatypes/<id> HTTP/1.1
Host: DOMAIN

# Response
HTTP/1.1 404 Not Found
```



Note

The request fails, if this datatypes is in use by datasets or attributes in the domain.

Populating an HDF5 Domain

In this section, we put the interface "to work". We show a fictive request/response exchange between an HTTP client and an HDF5/REST service. The task is to reproduce the example listed in Appendix B, *Example.h5*.

We assume that an HDF5 domain has been created at the URL `DOMAIN` with the root group at URI

```
DOMAIN/groups/903d1d75-e617-4767-a3bf-0cb3ee509027
```

(For brevity, some HTTP headers are not shown.)

```

1 #####
# Request to create attribute 'attr1' of the root group
PUT DOMAIN/groups/903d1d75-e617-4767-a3bf-0cb3ee509027/attributes/attr1 HTTP/1.1
Content-Type: application/json
5
{
  "type": {
    "class": "H5T_STRING",
    "length": 17,
10    "strPad": "H5T_STR_NULLTERM",
    "charSet": "H5T_CSET_ASCII"
  }
  "shape": "H5S_SCALAR",
  "value": "string attribute"
15 }

# Response
HTTP/1.1 201 Created
Location: DOMAIN/groups/903d1d75-e617-4767-a3bf-0cb3ee509027/attributes/attr1
20 Content-Type: application/json

{
  "links": {
    { "rel": "owner",
25    "href" : "DOMAIN/group/903d1d75-e617-4767-a3bf-0cb3ee509027" },
    { "rel": "root",
      "href" : "DOMAIN/root" },
    { "rel": "self",
      "href": "DOMAIN/group/903d1d75-e617-4767-a3bf-0cb3ee509027/attributes/attr1" }
30  }
}

#####
# Request to create a new (unlinked) dataset
35 POST DOMAIN/datasets HTTP/1.1
Content-Type: application/json

{
  "type": "H5T_STD_I32BE",
40  "shape": [10, 10],
  "value": [
    [0,1,2,3,4,5,6,7,8,9],
    [0,1,2,3,4,5,6,7,8,9],
    [0,1,2,3,4,5,6,7,8,9],
45  [0,1,2,3,4,5,6,7,8,9],
    [0,1,2,3,4,5,6,7,8,9],

```

```

    [0,1,2,3,4,5,6,7,8,9],
    [0,1,2,3,4,5,6,7,8,9],
    [0,1,2,3,4,5,6,7,8,9],
50    [0,1,2,3,4,5,6,7,8,9],
    [0,1,2,3,4,5,6,7,8,9]
  ]
}

55 # Response
HTTP/1.1 201 Created
Location: DOMAIN/datasets/30292613-8d2a-4dc4-a277-b9d59d5b0d20
Content-Type: application/json

60 {
  "id": "30292613-8d2a-4dc4-a277-b9d59d5b0d20",

  "links": [
    { "rel": "attributes",
65     "href" : "DOMAIN/datasets/30292613-8d2a-4dc4-a277-b9d59d5b0d20/attributes" },
    { "rel": "self",
      "href" : "DOMAIN/datasets/30292613-8d2a-4dc4-a277-b9d59d5b0d20" },
    { "rel": "root",
70     "href" : "DOMAIN/root" },
  ],

  "link-templates": [
    {
75     "rel": "participant",
      "href": "DOMAIN/groups/{id}/participants/{name}",
      "method": "PUT",
      "title": "Link to a group"
    }
  ]
80 }

#####
# Request to link the new dataset as 'dset1' in the root group
PUT DOMAIN/groups/903d1d75-e617-4767-a3bf-0cb3ee509027/participants/dset1 HTTP/1.1
85 Content-Type: application/json

{
  "idref": "30292613-8d2a-4dc4-a277-b9d59d5b0d20"
}
90 # Response
HTTP/1.1 201 Created
Location: DOMAIN/groups/903d1d75-e617-4767-a3bf-0cb3ee509027/participants/dset1
Content-Type: application/json
95 {
  "title": "dset1",
  "idref": "30292613-8d2a-4dc4-a277-b9d59d5b0d20"

100 "links": [
  { "rel": "group",
    "href" : "DOMAIN/groups/903d1d75-e617-4767-a3bf-0cb3ee509027" },
  { "rel": "participants",
    "href" : "DOMAIN/groups/903d1d75-e617-4767-a3bf-0cb3ee509027/participants" },
105 { "rel": "self",
    "href" : "DOMAIN/groups/903d1d75-e617-4767-a3bf-0cb3ee509027/participants/dset1" },

```

```
        { "rel": "root",
          "href" : "DOMAIN/root" },
    ]
110 }

#####
# Request to create a new (unlinked) dataset
POST DOMAIN/datasets HTTP/1.1
115 Content-Type: application/json

{
  "type" : {
    "class": "H5T_COMPOUND",
120     "members" : {
        "a": "H5T_STD_I32BE",
        "b": "H5T_IEEE_F32BE",
        "c": "H5T_IEEE_F64BE"
    }
125 },
  "shape": [5],
  "value": [
    { "a": 1, "b": 0.1, "c": 0.01 },
    { "a": 2, "b": 0.2, "c": 0.02 },
130     { "a": 3, "b": 0.3, "c": 0.03 },
    { "a": 4, "b": 0.4, "c": 0.04 },
    { "a": 5, "b": 0.5, "c": 0.05 }
  ]
}
135
# Response
HTTP/1.1 201 Created
Location: DOMAIN/datasets/0a68caca-629a-44aa-9f37-311e7ffb8417
Content-Type: application/json
140
{
  "id": "0a68caca-629a-44aa-9f37-311e7ffb8417",

  "links": [
145     { "rel": "attributes",
        "href" : "DOMAIN/datasets/0a68caca-629a-44aa-9f37-311e7ffb8417/attributes" },
        { "rel": "self",
          "href" : "DOMAIN/datasets/0a68caca-629a-44aa-9f37-311e7ffb8417" },
        { "rel": "root",
150         "href" : "DOMAIN/root" },
    ],

  "link-templates": [
    {
155     "rel": "participant",
      "href": "DOMAIN/groups/{id}/participants/{name}",
      "method": "PUT",
      "title": "Link to a group"
    }
160 ]
}

#####
# Request to link the new dataset as 'dset2' in the root group
165 PUT DOMAIN/groups/903d1d75-e617-4767-a3bf-0cb3ee509027/participants/dset2 HTTP/1.1
Content-Type: application/json
```

```

    {
      "idref": "0a68caca-629a-44aa-9f37-311e7ffb8417"
170 }

# Response
HTTP/1.1 201 Created
Location: DOMAIN/groups/903d1d75-e617-4767-a3bf-0cb3ee509027/participants/dset2
175 Content-Type: application/json

    {
      "title": "dset2",
      "idref": "0a68caca-629a-44aa-9f37-311e7ffb8417"
180
      "links": [
        { "rel": "group",
          "href" : "DOMAIN/groups/903d1d75-e617-4767-a3bf-0cb3ee509027" },
        { "rel": "participants",
185       "href" : "DOMAIN/groups/903d1d75-e617-4767-a3bf-0cb3ee509027/participants" },
        { "rel": "self",
          "href" : "DOMAIN/groups/903d1d75-e617-4767-a3bf-0cb3ee509027/participants/dset2" },
        { "rel": "root",
          "href" : "DOMAIN/root" }
190     ]
    }

#####
# Request to create a new (unlinked) group
195 POST DOMAIN/groups HTTP/1.1
Content-Type: application/json

# Response
HTTP/1.1 201 Created
200 Location: DOMAIN/groups/be8dcb22-b411-4439-85e9-ea384a685ae0
Content-Type: application/json

    {
      "id": "be8dcb22-b411-4439-85e9-ea384a685ae0",
205
      "links": [
        { "rel": "attributes",
          "href" : "DOMAIN/groups/be8dcb22-b411-4439-85e9-ea384a685ae0/attributes" },
        { "rel": "participants",
210       "href" : "DOMAIN/groups/be8dcb22-b411-4439-85e9-ea384a685ae0/participants" },
        { "rel": "self",
          "href" : "DOMAIN/groups/be8dcb22-b411-4439-85e9-ea384a685ae0" },
        { "rel": "root",
          "href" : "DOMAIN/root" }
215     ],
      "link-templates": [
        {
          "rel": "participant",
220       "href": "DOMAIN/groups/{id}/participants/{name}",
          "method": "PUT",
          "title": "Link to a group"
        }
      ]
225 }

```



```
#####
# Request to link the new group as 'group1' in the root group
PUT DOMAIN/groups/903d1d75-e617-4767-a3bf-0cb3ee509027/participants/group1 HTTP/1.1
230 Content-Type: application/json

{
  "idref": "be8dcb22-b411-4439-85e9-ea384a685ae0"
}
235
# Response
HTTP/1.1 201 Created
Location: DOMAIN/groups/903d1d75-e617-4767-a3bf-0cb3ee509027/participants/group1
Content-Type: application/json
240
{
  "title": "group1",
  "idref": "be8dcb22-b411-4439-85e9-ea384a685ae0"

245  "links": [
    { "rel": "group",
      "href" : "DOMAIN/groups/903d1d75-e617-4767-a3bf-0cb3ee509027" },
    { "rel": "participants",
      "href" : "DOMAIN/groups/903d1d75-e617-4767-a3bf-0cb3ee509027/participants" },
250  { "rel": "self",
      "href" : "DOMAIN/groups/903d1d75-e617-4767-a3bf-0cb3ee509027/participants/group1" },
    { "rel": "root",
      "href" : "DOMAIN/root" }
  ]
255 }

#####
# Request to create a new (unlinked) datatype
POST DOMAIN/datatypes HTTP/1.1
260 Content-Type: application/json

{
  "type": {
    "class": "H5T_COMPOUND",
265    "members": {
      "a": {
        "class": "H5T_ARRAY",
        "base": "H5T_STD_I32BE",
        "dim": [4]
270      },
      "b": {
        "class": "H5T_ARRAY",
        "base": "H5T_IEEE_F32BE",
        "dim": [5, 6]
275      }
    }
  }
}

280 # Response
HTTP/1.1 201 Created
Location: DOMAIN/datatypes/a93ff089-d466-44e7-b3f0-09db34ec2ef5
Content-Type: application/json

285 {
  "id": "a93ff089-d466-44e7-b3f0-09db34ec2ef5",
```

```

    "links": [
290     { "rel": "attributes",
        "href" : "DOMAIN/datatypes/a93ff089-d466-44e7-b3f0-09db34ec2ef5/attributes" },
        { "rel": "self",
          "href" : "DOMAIN/datatypes/a93ff089-d466-44e7-b3f0-09db34ec2ef5" },
        { "rel": "root",
          "href" : "DOMAIN/root" }
295   ],

    "link-templates": [
        {
300     "rel": "participant",
        "href": "DOMAIN/groups/{id}/participants/{name}",
        "method": "PUT",
        "title": "Link to a group"
        }
    ]
305 }

#####
# Request to link the new datatype as 'type1' in the root group
PUT DOMAIN/groups/903d1d75-e617-4767-a3bf-0cb3ee509027/participants/type1 HTTP/1.1
310 Content-Type: application/json

{
    "idref": "a93ff089-d466-44e7-b3f0-09db34ec2ef5"
}
315
# Response
HTTP/1.1 201 Created
Location: DOMAIN/groups/903d1d75-e617-4767-a3bf-0cb3ee509027/participants/type1
Content-Type: application/json
320
{
    "title": "type1",
    "idref": "a93ff089-d466-44e7-b3f0-09db34ec2ef5"

325    "links": [
        { "rel": "group",
          "href" : "DOMAIN/groups/903d1d75-e617-4767-a3bf-0cb3ee509027" },
        { "rel": "participants",
          "href" : "DOMAIN/groups/903d1d75-e617-4767-a3bf-0cb3ee509027/participants" },
330        { "rel": "self",
          "href" : "DOMAIN/groups/903d1d75-e617-4767-a3bf-0cb3ee509027/participants/type1" },
        { "rel": "root",
          "href" : "DOMAIN/root" }
    ]
335 }

#####
# Request to create a new (unlinked) dataset (uses '/type1')
POST DOMAIN/datasets HTTP/1.1
340 Content-Type: application/json

{
    "type": {
345     "hdf5": "/type1"
    },
    "shape": [5],

```

```
"value": [  
  {  
    "a": [0,1,2,3],  
    "b": [  
350      [0.1,0.1,0.1,0.1,0.1,0.1],  
        [0.2,0.2,0.2,0.2,0.2,0.2],  
        [0.3,0.3,0.3,0.3,0.3,0.3],  
        [0.4,0.4,0.4,0.4,0.4,0.4],  
355      [0.5,0.5,0.5,0.5,0.5,0.5]  
    ]  
  },  
  {  
    "a": [0,1,2,3],  
    "b": [  
360      [0.1,0.1,0.1,0.1,0.1,0.1],  
        [0.2,0.2,0.2,0.2,0.2,0.2],  
        [0.3,0.3,0.3,0.3,0.3,0.3],  
        [0.4,0.4,0.4,0.4,0.4,0.4],  
365      [0.5,0.5,0.5,0.5,0.5,0.5]  
    ]  
  },  
  {  
    "a": [0,1,2,3],  
    "b": [  
370      [0.1,0.1,0.1,0.1,0.1,0.1],  
        [0.2,0.2,0.2,0.2,0.2,0.2],  
        [0.3,0.3,0.3,0.3,0.3,0.3],  
        [0.4,0.4,0.4,0.4,0.4,0.4],  
375      [0.5,0.5,0.5,0.5,0.5,0.5]  
    ]  
  },  
  {  
    "a": [0,1,2,3],  
    "b": [  
380      [0.1,0.1,0.1,0.1,0.1,0.1],  
        [0.2,0.2,0.2,0.2,0.2,0.2],  
        [0.3,0.3,0.3,0.3,0.3,0.3],  
        [0.4,0.4,0.4,0.4,0.4,0.4],  
385      [0.5,0.5,0.5,0.5,0.5,0.5]  
    ]  
  },  
  {  
    "a": [0,1,2,3],  
    "b": [  
390      [0.1,0.1,0.1,0.1,0.1,0.1],  
        [0.2,0.2,0.2,0.2,0.2,0.2],  
        [0.3,0.3,0.3,0.3,0.3,0.3],  
        [0.4,0.4,0.4,0.4,0.4,0.4],  
395      [0.5,0.5,0.5,0.5,0.5,0.5]  
    ]  
  }  
]  
}  
400 # Response  
    HTTP/1.1 201 Created  
    Location: DOMAIN/datasets/42f5e3a2-5e70-4faf-9893-fd216257a0d9  
    Content-Type: application/json  
405 {  
  {
```

```

    "id": "42f5e3a2-5e70-4faf-9893-fd216257a0d9",
    "links": [
410     { "rel": "attributes",
        "href" : "DOMAIN/datasets/42f5e3a2-5e70-4faf-9893-fd216257a0d9/attributes" },
        { "rel": "self",
          "href" : "DOMAIN/datasets/42f5e3a2-5e70-4faf-9893-fd216257a0d9" },
        { "rel": "root",
          "href" : "DOMAIN/root" },
415     ],
    "link-templates": [
        {
420     "rel": "participant",
        "href": "DOMAIN/groups/{id}/participants/{name}",
        "method": "PUT",
        "title": "Link to a group"
        }
425     ]
    }

#####
# Request to link the new dataset as 'dset3' in the group '/group1'
430 PUT DOMAIN/groups/be8dcb22-b411-4439-85e9-ea384a685ae0/participants/dset3 HTTP/1.1
Content-Type: application/json

    {
        "idref": "42f5e3a2-5e70-4faf-9893-fd216257a0d9"
435     }

# Response
HTTP/1.1 201 Created
Location: DOMAIN/groups/be8dcb22-b411-4439-85e9-ea384a685ae0/participants/dset3
440 Content-Type: application/json

    {
        "title": "dset3",
        "idref": "42f5e3a2-5e70-4faf-9893-fd216257a0d9"
445     "links": [
        { "rel": "group",
          "href" : "DOMAIN/groups/be8dcb22-b411-4439-85e9-ea384a685ae0" },
        { "rel": "participants",
          "href" : "DOMAIN/groups/be8dcb22-b411-4439-85e9-ea384a685ae0/participants" },
450     { "rel": "self",
          "href" : "DOMAIN/groups/be8dcb22-b411-4439-85e9-ea384a685ae0/participants/dset3" },
        { "rel": "root",
          "href" : "DOMAIN/root" }
455     ]
    }

#####
# Request to create a new (unlinked) dataset
460 POST DOMAIN/datasets HTTP/1.1
Content-Type: application/json

    {
        "type": {
465     "class": "H5T_VLEN",
        "base": "H5T_STD_I32LE"
    }

```

```
    },
    "shape": [4],
    "value": [
470     [0], [10 11], [20 21 22], [30 31 32 33]
    ]
  }

# Response
475 HTTP/1.1 201 Created
Location: DOMAIN/datasets/4b43748e-817f-44c6-a9f1-16e242fd374b
Content-Type: application/json

{
480   "id": "4b43748e-817f-44c6-a9f1-16e242fd374b",

   "links": [
     { "rel": "attributes",
485       "href" : "DOMAIN/datasets/4b43748e-817f-44c6-a9f1-16e242fd374b/attributes" },
     { "rel": "self",
       "href" : "DOMAIN/datasets/4b43748e-817f-44c6-a9f1-16e242fd374b" },
     { "rel": "root",
       "href" : "DOMAIN/root" },
490   ],

   "link-templates": [
     {
495       "rel": "participant",
       "href": "DOMAIN/groups/{id}/participants/{name}",
       "method": "PUT",
       "title": "Link to a group"
     }
   ]
}
500
#####
# Request to link the new dataset as 'dset3' in the root group
PUT DOMAIN/groups/903d1d75-e617-4767-a3bf-0cb3ee509027/participants/dset3 HTTP/1.1
Content-Type: application/json
505
{
  "idref": "4b43748e-817f-44c6-a9f1-16e242fd374b"
}

510 # Response
HTTP/1.1 201 Created
Location: DOMAIN/groups/903d1d75-e617-4767-a3bf-0cb3ee509027/participants/dset3
Content-Type: application/json

515 {
  "title": "dset3",
  "idref": "4b43748e-817f-44c6-a9f1-16e242fd374b"

  "links": [
520    { "rel": "group",
      "href" : "DOMAIN/groups/903d1d75-e617-4767-a3bf-0cb3ee509027" },
    { "rel": "participants",
      "href" : "DOMAIN/groups/903d1d75-e617-4767-a3bf-0cb3ee509027/participants" },
    { "rel": "self",
525      "href" : "DOMAIN/groups/903d1d75-e617-4767-a3bf-0cb3ee509027/participants/dset3" },
    { "rel": "root",
```

```

        "href" : "DOMAIN/root" }
    ]
}
530 #####
# Request to link the group at '/group1' as 'group2' in the root group
PUT DOMAIN/groups/903d1d75-e617-4767-a3bf-0cb3ee509027/participants/group2 HTTP/1.1
Content-Type: application/json
535 {
    "idref": "be8dcb22-b411-4439-85e9-ea384a685ae0"
}
540 # Response
HTTP/1.1 201 Created
Location: DOMAIN/groups/903d1d75-e617-4767-a3bf-0cb3ee509027/participants/group2
Content-Type: application/json
545 {
    "title": "group2",
    "idref": "be8dcb22-b411-4439-85e9-ea384a685ae0"

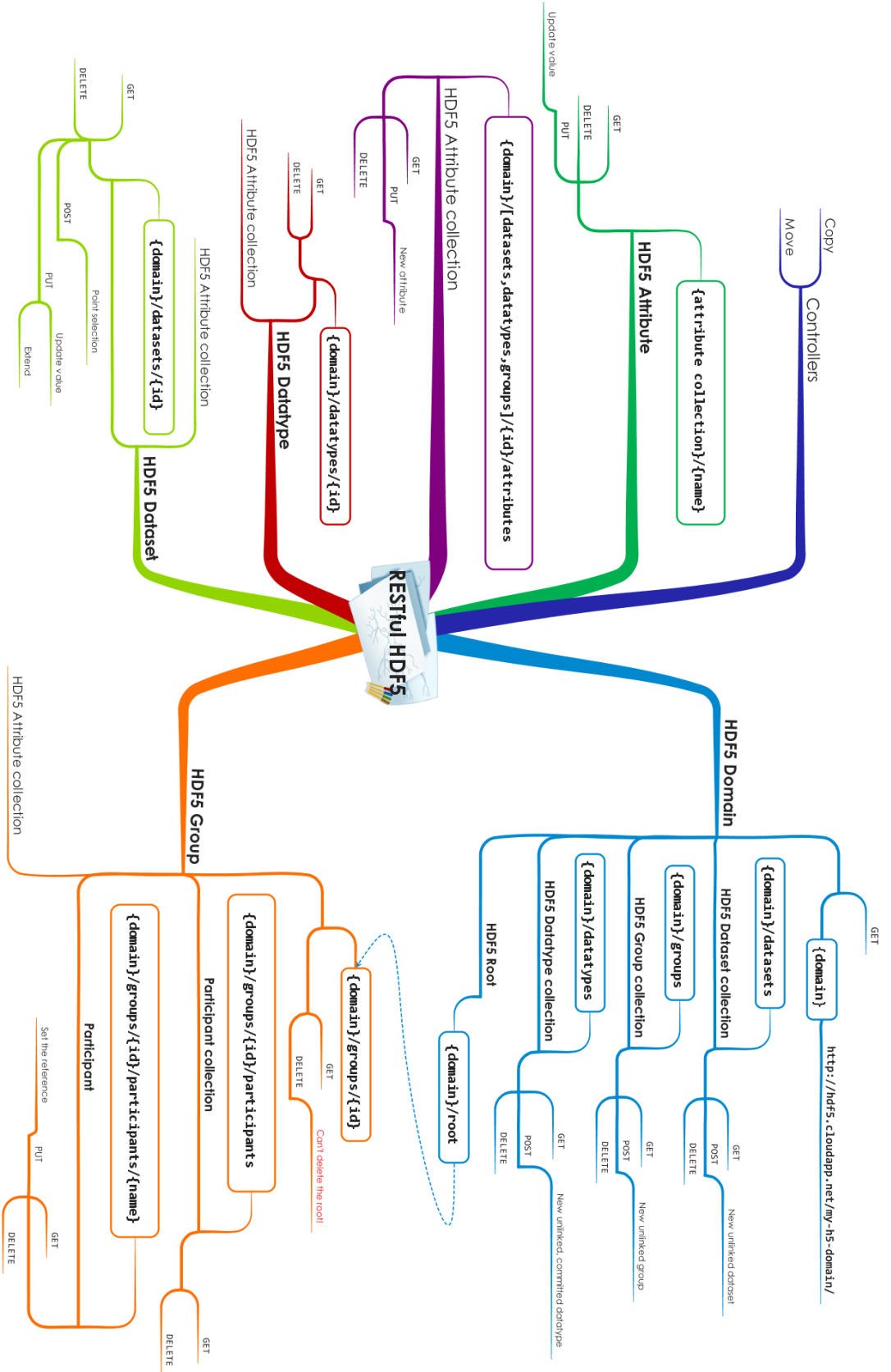
    "links": [
550     { "rel": "group",
        "href" : "DOMAIN/groups/903d1d75-e617-4767-a3bf-0cb3ee509027" },
        { "rel": "participants",
          "href" : "DOMAIN/groups/903d1d75-e617-4767-a3bf-0cb3ee509027/participants" },
555     { "rel": "self",
        "href" : "DOMAIN/groups/903d1d75-e617-4767-a3bf-0cb3ee509027/participants/group2" },
        { "rel": "root",
          "href" : "DOMAIN/root" }
    ]
}
560 #####
# Request to create a link 'slink1' in the root group
PUT DOMAIN/groups/903d1d75-e617-4767-a3bf-0cb3ee509027/participants/slink1 HTTP/1.1
Content-Type: application/json
565 {
    "hdf5": "somevalue"
}
570 # Response
HTTP/1.1 201 Created
Location: DOMAIN/groups/903d1d75-e617-4767-a3bf-0cb3ee509027/participants/slink1
Content-Type: application/json
575 {
    "title": "slink1",
    "hdf5": "somevalue"

    "links": [
580     { "rel": "group",
        "href" : "DOMAIN/groups/903d1d75-e617-4767-a3bf-0cb3ee509027" },
        { "rel": "participants",
          "href" : "DOMAIN/groups/903d1d75-e617-4767-a3bf-0cb3ee509027/participants" },
585     { "rel": "self",
        "href" : "DOMAIN/groups/903d1d75-e617-4767-a3bf-0cb3ee509027/participants/slink1" },
        { "rel": "root",
          "href" : "DOMAIN/root" }
    ]
}

```

```
    "href" : "DOMAIN/root" }  
  ]  
}  
590 #####
```

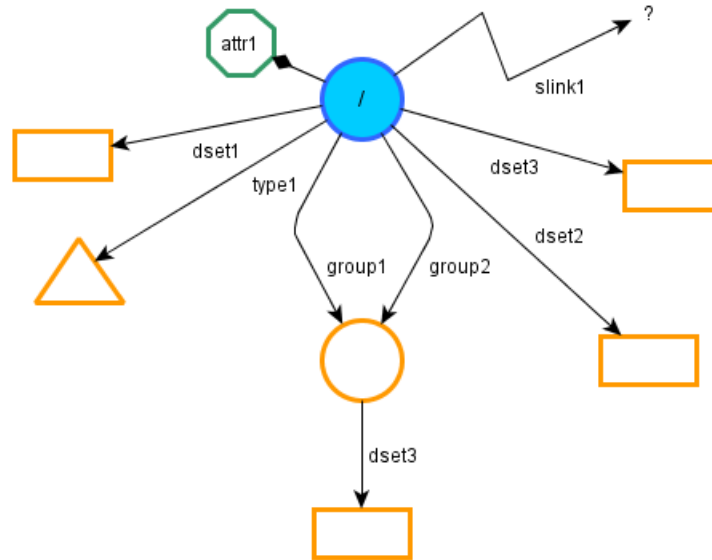
A. RESTful HDF5 Overview



B. Example.h5

Throughout this document we've used a standard HDF5 example from the HDF5 documentation [BNFDDL]. In the figure below, a multigraph representation of our example is shown. Circles represent HDF5 groups, rectangles represent HDF5 datasets, triangles represent HDF5 datatypes, hexagons represent HDF5 attributes, and (labelled) arrows represent HDF5 links. There are two groups, four datasets, one linked datatype, and a soft link. The root group (blue circle) has an attribute.

Figure B.1. Infoset Multigraph of Example.h5



Circles represent HDF5 groups, rectangles represent HDF5 datasets, triangles represent HDF5 datatypes, hexagons represent HDF5 attributes, and (labelled) arrows represent HDF5 links. There are two groups, four datasets, one linked datatype, and a soft link. The root group (blue circle) has one attribute. Note that the non-root group is linked twice under different names, i.e., the path names `/group1` and `/group2` lead to the same group.

Below, the output of running `h5dump` against `Example.h5` is shown.

```

1 HDF5 "Example.h5" {
  GROUP "/" {
    ATTRIBUTE "attr1" {
      DATATYPE H5T_STRING {
5         STRSIZE 17;
          STRPAD H5T_STR_NULLTERM;
          CSET H5T_CSET_ASCII;
          CTYPE H5T_C_S1;
      }
10    DATASPACE SCALAR
      DATA {
        "string attribute"
      }
    }
15  DATASET "dset1" {
    DATATYPE H5T_STD_I32BE
    DATASPACE SIMPLE { ( 10, 10 ) / ( 10, 10 ) }
  }
  DATASET "dset2" {
    DATATYPE H5T_STD_I32BE
    DATASPACE SIMPLE { ( 10, 10 ) / ( 10, 10 ) }
  }
  DATASET "dset3" {
    DATATYPE H5T_STD_I32BE
    DATASPACE SIMPLE { ( 10, 10 ) / ( 10, 10 ) }
  }
  DATASET "dset3" {
    DATATYPE H5T_STD_I32BE
    DATASPACE SIMPLE { ( 10, 10 ) / ( 10, 10 ) }
  }
  DATATYPE "type1" {
    DATATYPE H5T_STRING {
      STRSIZE 17;
      STRPAD H5T_STR_NULLTERM;
      CSET H5T_CSET_ASCII;
      CTYPE H5T_C_S1;
    }
  }
  GROUP "group1" {
    DATASET "dset3" {
      DATATYPE H5T_STD_I32BE
      DATASPACE SIMPLE { ( 10, 10 ) / ( 10, 10 ) }
    }
  }
  GROUP "group2" {
    DATASET "dset3" {
      DATATYPE H5T_STD_I32BE
      DATASPACE SIMPLE { ( 10, 10 ) / ( 10, 10 ) }
    }
  }
  SOFT_LINK "slink1" {
    ?
  }
}

```

```

DATA {
  0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
20  0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
    0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
    0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
    0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
    0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
25  0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
    0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
    0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
    0, 1, 2, 3, 4, 5, 6, 7, 8, 9
}
30 }
DATASET "dset2" {
  DATATYPE H5T_COMPOUND {
    H5T_STD_I32BE "a";
    H5T_IEEE_F32BE "b";
35  H5T_IEEE_F64BE "c";
  }
  DATASPACE SIMPLE { ( 5 ) / ( 5 ) }
  DATA {
40  { 1, 0.1, 0.01 },
    { 2, 0.2, 0.02 },
    { 3, 0.3, 0.03 },
    { 4, 0.4, 0.04 },
    { 5, 0.5, 0.05 }
  }
45 }
GROUP "group1" {
  DATASET "dset3" {
    DATATYPE "/type1"
    DATASPACE SIMPLE { ( 5 ) / ( 5 ) }
50  DATA {
    {
      [ 0, 1, 2, 3 ],
      [ 0.1, 0.1, 0.1, 0.1, 0.1, 0.1,
55  0.2, 0.2, 0.2, 0.2, 0.2, 0.2,
        0.3, 0.3, 0.3, 0.3, 0.3, 0.3,
        0.4, 0.4, 0.4, 0.4, 0.4, 0.4,
        0.5, 0.5, 0.5, 0.5, 0.5, 0.5 ]
    },
    {
60  [ 0, 1, 2, 3 ],
      [ 0.1, 0.1, 0.1, 0.1, 0.1, 0.1,
        0.2, 0.2, 0.2, 0.2, 0.2, 0.2,
        0.3, 0.3, 0.3, 0.3, 0.3, 0.3,
        0.4, 0.4, 0.4, 0.4, 0.4, 0.4,
65  0.5, 0.5, 0.5, 0.5, 0.5, 0.5 ]
    },
    {
      [ 0, 1, 2, 3 ],
      [ 0.1, 0.1, 0.1, 0.1, 0.1, 0.1,
70  0.2, 0.2, 0.2, 0.2, 0.2, 0.2,
        0.3, 0.3, 0.3, 0.3, 0.3, 0.3,
        0.4, 0.4, 0.4, 0.4, 0.4, 0.4,
        0.5, 0.5, 0.5, 0.5, 0.5, 0.5 ]
    },
75  {
      [ 0, 1, 2, 3 ],
      [ 0.1, 0.1, 0.1, 0.1, 0.1, 0.1,

```

```

80         0.2, 0.2, 0.2, 0.2, 0.2, 0.2,
           0.3, 0.3, 0.3, 0.3, 0.3, 0.3,
           0.4, 0.4, 0.4, 0.4, 0.4, 0.4,
           0.5, 0.5, 0.5, 0.5, 0.5, 0.5 ]
           },
           {
85         [ 0, 1, 2, 3 ],
           [ 0.1, 0.1, 0.1, 0.1, 0.1, 0.1,
             0.2, 0.2, 0.2, 0.2, 0.2, 0.2,
             0.3, 0.3, 0.3, 0.3, 0.3, 0.3,
             0.4, 0.4, 0.4, 0.4, 0.4, 0.4,
             0.5, 0.5, 0.5, 0.5, 0.5, 0.5 ]
90         }
           }
         }
       }
     DATASET "dset3" {
95       DATATYPE H5T_VLEN { H5T_STD_I32LE }
       DATASPACE SIMPLE { ( 4 ) / ( 4 ) }
       DATA {
         (0), (10, 11), (20, 21, 22), (30, 31, 32, 33)
100      }
     }
     GROUP "group2" {
       HARDLINK "/group1"
     }
     SOFTLINK "slink1" {
105     LINKTARGET "somevalue"
     }
     DATATYPE "type1" H5T_COMPOUND {
       H5T_ARRAY { [4] H5T_STD_I32BE } "a";
       H5T_ARRAY { [5][6] H5T_IEEE_F32BE } "b";
110   }
   }
 }

```

C. HDF5/JSON

BNF Grammar

```

1 ;=====
  ; tokens in alphabetical order
  ;=====

5 <array> ::= "{"
           "class" ":" "H5T_ARRAY" ", "
           "base"  ":" <integer>  ", "
           "dim"   ":" <dim>
           "}"

10 <attribute> ::= "{"
           "name" ":" <string_value>  ", "
           "type" ":" <datatype>      ", "
           "shape" ":" <dataspace>    ", "
15          "value" ":" <attribute_value>
           "}"

```

```

<attribute_collection> ::= <attribute_sans_val> "," <attribute_collection> |
  <attribute_sans_val>
20 <attribute_sans_val> ::= "{"
    "name" ":" <string_value> ","
    "type" ":" <datatype> ","
    "shape" ":" <dataspace>
25 "}"

<attribute_value> ::= <value>

<bitfield> ::= <bitfield_user> | <bitfield_pre>
30 <bitfield_pre> ::= "H5T_STD_B8BE" | "H5T_STD_B8LE" |
    "H5T_STD_B16BE" | "H5T_STD_B16LE" |
    "H5T_STD_B32BE" | "H5T_STD_B32LE" |
    "H5T_STD_B64BE" | "H5T_STD_B64LE"
35 <bitfield_user> ::= "{"
    "bitOffset" ":" <non_neg_int_value> ","
    "byteOrder" ":" <byte_order> ","
    "class" ":" "H5T_BITFIELD" ","
40 "lsbPad" ":" "<bit_padding>" ","
    "msbPad" ":" "<bit_padding>" ","
    "precision" ":" "<pos_int_value>" ","
    "size" ":" "<pos_int_value>"
    "}"
45 <bit_padding> ::= "H5T_PAD_ZERO" | "H5T_PAD_ONE" | "H5T_PAD_BACKGROUND"

<byte_order> ::= "H5T_ORDER_LE" | "H5T_ORDER_BE"

50 <compound> ::= "{"
    "class" ":" "H5T_COMPOUND" ","
    "members" ":" "{" <compound_member_collection> "}"
    "}"

55 <compound_member> ::= <string_value> ":" <datatype>

<compound_member_collection> ::=
  <compound_member> "," <compound_member_collection> | <compound_member>

60 <dataset> ::= "{"
    "id" ":" <id> ","
    [ "attributes" ":" <attribute_collection> "," ]
    "type" ":" <datatype> ","
    "shape" ":" <dataspace> ","
65 "value" ":" <attribute_value>
    "}"

<dataset_collection> ::= <dataset_sans_val>,< dataset_collection> |
  <dataset_sans_val>
70 <dataset_sans_val> ::= "{"
    "id" ":" <id> ","
    [ "attributes" ":" <attribute_collection> "," ]
    "type" ":" <datatype> ","
75 "shape" ":" <dataspace>
    "}"

```

```

<dataset_value> ::= <value>
80 <dataspace> ::= "H5S_NULL" | "H5S_SCALAR" | <dim> | <simple_dataspace>

<datatype> ::= <array> | <bitfield> | <compound> | <enum> | <float> |
               <integer> | <opaque> | <reference> | <string> | <vlen>

85 <datatype_collection> ::= <datatype>, <datatype_collection> | <datatype>

<dim> ::= <non_neg_int_value> | <non_neg_int_value>, <dim>

<enum> ::= "{"
90   "class"   ":" "H5T_ENUM"           ","
   "base"    ":" <integer>           ","
   "members" ":" <enum_member_collection>
   "}"

95 <enum_member> ::= <string_value> ":" <int_value>

<enum_member_collection> ::=
   <enum_member> ", " <enum_member_collection> | <enum_member>

100 <float> ::= <float_pre> | <float_user>

<float_pre> ::= "H5T_IEEE_F32BE" | "H5T_IEEE_F32LE" |
               "H5T_IEEE_F64BE" | "H5T_IEEE_F64LE"

105 <float_user> ::= "{"
   "bitOffset"   ":" <non_neg_int_value> ","
   "byteOrder"  ":" <byte_order>         ","
   "class"       ":" "H5T_FLOAT"         ","
   "expBias"     ":" <pos_int_value>     ","
110  "expBits"   ":" <pos_int_value>     ","
   "expBitPos"  ":" <pos_int_value>     ","
   "intlbPad"   ":" <bit_padding>       ","
   "lsbPad"     ":" <bit_padding>       ","
   "mantBits"   ":" <pos_int_value>     ","
115  "mantBitPos" ":" <non_neg_int_value> ","
   "mantNorm"   ":" <mant_norm>         ","
   "msbitPad"   ":" <bit_padding>       ","
   "precision"  ":" <pos_int_value>     ","
   "signBitPos" ":" <pos_int_value>     ","
120  "size"     ":" <pos_int_value>
   "}"

<group> ::= {
125  "id"         ":" <id>               ","
   "attributes" ":" <attribute_collection> ","
   "participants" ":" <participant_collection>
   }

130 <group_collection> ::= <group>, <group_collection> | <group>

<h5path> ::= an absolute or relative HDF5 path name

<hyperslab_set> ::= "H5S_SELECT_SET" ":" <simple_hyperslab>
135 <hyperslabs> ::= <hyperslab_set> | <hyperslabs_and> | <hyperslabs_not> |

```

```

    <hyperslabs_notb> | <hyperslabs_or> | <hyperslabs_or> | <hyperslabs_xor>

<hyperslabs_and> ::= "H5S_SELECT_AND" ":" "["
140 <simple_hyperslab> "," <hyperslabs>
    "]"

<hyperslabs_nota> ::= "H5S_SELECT_NOTA" ":" "["
145 <simple_hyperslab> "," <hyperslabs>
    "]"

<hyperslabs_notb> ::= "H5S_SELECT_NOTB" ":" "["
    <simple_hyperslab> "," <hyperslabs>
    "]"
150

<hyperslabs_or> ::= "H5S_SELECT_OR" ":" "["
    <simple_hyperslab> "," <hyperslabs>
    "]"

155 <hyperslabs_xor> ::= "H5S_SELECT_XOR" ":" "["
    <simple_hyperslab> "," <hyperslabs>
    "]"

<id> ::= /[a-f0-9]{8}-[a-f0-9]{4}-[a-f0-9]{4}-[a-f0-9]{4}-[a-f0-9]{12}/
160
<idref> ::= <id>

<integer> ::= <integer_pre> | <integer_user>

165 <integer_pre> ::= "H5T_STD_I8BE" | "H5T_STD_I8LE" |
    "H5T_STD_I16BE" | "H5T_STD_I16LE" |
    "H5T_STD_I32BE" | "H5T_STD_I32LE" |
    "H5T_STD_I64BE" | "H5T_STD_I64LE" |
    "H5T_STD_U8BE" | "H5T_STD_U8LE" |
170 "H5T_STD_U16BE" | "H5T_STD_U16LE" |
    "H5T_STD_U32BE" | "H5T_STD_U32LE" |
    "H5T_STD_U64BE" | "H5T_STD_U64LE"

<integer_user> ::= "{"
175 "bitOffset" ":" <non_neg_int_value> ","
    "byteOrder" ":" <byte_order> ","
    "class" ":" "H5T_INTEGER" ","
    "lsbPad" ":" <bit_padding> ","
    "msbPad" ":" <bit_padding> ","
180 "precision" ":" <pos_int_value> ","
    "signType" ":" <sign_type> ","
    "size" ":" <pos_int_value>
    "}"

185 <locator> ::= <idref> | <h5path> | <url>

<mant_norm> ::= "H5T_NORM IMPLIED" | "H5T_NORM_MSBSET" | "H5T_NORM_NONE"

<max_dim> ::= <max_dim_value> | <max_dim_value>, <max_dim>
190
<max_dim_value> ::= <non_neg_int_value> | "H5S_UNLIMITED"

<opaque> ::= "{"
    "class" ":" "H5T_OPAQUE" ","
195 [ "tag" ":" <string_value> "," ]
    "size" ":" <pos_int_value>

```

```

    }"
    <reference> ::= "H5T_STD_REF_OBJECT" | "H5T_STD_REF_DSETREG"
200 <participant> ::= "{"
    "title": <string_value> ","
    "idref" ":" <id> | "hdf5" ":" <HDF5 path name> | "href" ":" <url>
    }"
205 <participant_collection> ::= <participant> "," <participant_collection> |
    <participant>

    <selection> ::= <hyperslabs> | <points>
210 <simple_hyperslab> ::= "{"
    "start" ":" "[" <dim> "]" ","
    "stride" ":" "[" <dim> "]" ","
    "count" ":" "[" <dim> "]" ","
215 "block" ":" "[" <dim> "]"
    }"

    <simple_dataspace> ::= "{"
    "class" ":" "H5S_SIMPLE" ","
220 "dim" ":" "{"
    "curr" ":" <dim> ","
    "max" ":" <max_dim>
    }"
    "chunk" ":" <dim>
225 }"

    <sign_type> ::= "H5T_SGN_NONE" | "H5T_SGN_2"

    <string> ::= "{"
230 "class" ":" "H5T_STRING"
    "charSet" ":" "H5T_CSET_ASCII" | "H5T_CSET_UTF8"
    "length" ":" <pos_int_value> | "H5T_VARIABLE"
    "strPad" ":" "H5T_STR_NULLTERM" | "H5T_STR_NULLPAD" | "H5T_STR_SPACEPAD"
    }"
235 <string_value> ::= a string

    <url> ::= <url_sans_frag> "#" <h5path>
240 <url_sans_frag> ::= URL without a URL fragment

    <utctime> ::= ISO time string

    <value> ::= JSON value minus true, false, null
245 <vlen> ::= "{"
    "class" ":" "H5T_VLEN"
    "base" ":" <datatype>
    }"
250

```

Example

Below, an HDF5/JSON rendering of `Example.h5` is shown.

```
1 {
  "id": "e203fee7-89b4-4216-894d-7aef0e3a199d",
  "created": "1985-04-12T23:20:50.52",
  "lastModified": "1996-12-19T16:39:57",
5  "root": "903d1d75-e617-4767-a3bf-0cb3ee509027",
  "groups": [
    {
      "id": "be8dcb22-b411-4439-85e9-ea384a685ae0"
10     "participants": [
        {
          "title": "dset3",
          "idref": "42f5e3a2-5e70-4faf-9893-fd216257a0d9"
        }
      ]
15    },
    {
      "id": "903d1d75-e617-4767-a3bf-0cb3ee509027",
      "attributes": [
20        {
          "name": "attr1",
          "type": {
            "class": "H5T_STRING",
            "length": 17,
            "strPad": "H5T_STR_NULLTERM",
25            "charSet": "H5T_CSET_ASCII"
          },
          "shape": "H5S_SCALAR",
          "value": "string attribute"
        }
      ],
30    "participants": [
      {
        "title": "dset1",
        "idref": "30292613-8d2a-4dc4-a277-b9d59d5b0d20"
35      },
      {
        "title": "group1",
        "idref": "be8dcb22-b411-4439-85e9-ea384a685ae0"
40      },
      {
        "title": "group2",
        "idref": "be8dcb22-b411-4439-85e9-ea384a685ae0"
      },
      {
45        "title": "dset2",
        "idref": "0a68caca-629a-44aa-9f37-311e7ffb8417"
      },
      {
        "title": "dset3",
50        "idref": "4b43748e-817f-44c6-a9f1-16e242fd374b"
      },
      {
        "title": "slink1",
        "hdf5": "somevalue"
55      },
      {
        "title": "type1",
        "idref": "a93ff089-d466-44e7-b3f0-09db34ec2ef5"
      }
    ]
  ]
}
```



```
60     ]
      }
    ],
    "datasets": [
      {
65         "id": "30292613-8d2a-4dc4-a277-b9d59d5b0d20",
          "type": "H5T_STD_I32BE",
          "shape": [ 10, 10 ],
          "value": [
70             [ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 ],
              [ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 ],
              [ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 ],
              [ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 ],
              [ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 ],
75             [ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 ],
              [ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 ],
              [ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 ],
              [ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 ]
            ],
80         "layout": "H5D_CONTIGUOUS"
      },
      {
        "id": "0a68caca-629a-44aa-9f37-311e7ffb8417",
65         "type": {
          "class": "H5T_COMPOUND",
          "members": {
85             "a": "H5T_STD_I32BE",
              "b": "H5T_IEEE_F32BE",
              "c": "H5T_IEEE_F64BE"
            }
          },
90         "shape": [5],
          "value": [
            { "a": 1, "b": 0.1, "c": 0.01 },
95             { "a": 2, "b": 0.2, "c": 0.02 },
              { "a": 3, "b": 0.3, "c": 0.03 },
              { "a": 4, "b": 0.4, "c": 0.04 },
              { "a": 5, "b": 0.5, "c": 0.05 }
            ],
100        "layout": "H5D_CONTIGUOUS"
      },
      {
        "id": "4b43748e-817f-44c6-a9f1-16e242fd374b",
105        "type": {
          "class": "H5T_VLEN",
          "base": "H5T_STD_I32LE"
        },
        "shape": [4],
110        "value": [
          [0],
          [ 10, 11 ],
          [ 20, 21, 22 ],
          [ 30, 31, 32, 33 ]
        ],
115        "layout": "H5D_CONTIGUOUS"
      },
      {
        "id": "42f5e3a2-5e70-4faf-9893-fd216257a0d9",
        "type": {
```

```

120         "hdf5": "/type1"
        },
        "shape": [5],
        "value": [
125             {
                "a": [ 0, 1, 2, 3 ],
                "b": [
130                     [ 0.1, 0.1, 0.1, 0.1, 0.1, 0.1 ],
                     [ 0.2, 0.2, 0.2, 0.2, 0.2, 0.2 ],
                     [ 0.3, 0.3, 0.3, 0.3, 0.3, 0.3 ],
                     [ 0.4, 0.4, 0.4, 0.4, 0.4, 0.4 ],
                     [ 0.5, 0.5, 0.5, 0.5, 0.5, 0.5 ]
                ]
            },
135             {
                "a": [ 0, 1, 2, 3 ],
                "b": [
140                     [ 0.1, 0.1, 0.1, 0.1, 0.1, 0.1 ],
                     [ 0.2, 0.2, 0.2, 0.2, 0.2, 0.2 ],
                     [ 0.3, 0.3, 0.3, 0.3, 0.3, 0.3 ],
                     [ 0.4, 0.4, 0.4, 0.4, 0.4, 0.4 ],
                     [ 0.5, 0.5, 0.5, 0.5, 0.5, 0.5 ]
                ]
            },
145             {
                "a": [ 0, 1, 2, 3 ],
                "b": [
150                     [ 0.1, 0.1, 0.1, 0.1, 0.1, 0.1 ],
                     [ 0.2, 0.2, 0.2, 0.2, 0.2, 0.2 ],
                     [ 0.3, 0.3, 0.3, 0.3, 0.3, 0.3 ],
                     [ 0.4, 0.4, 0.4, 0.4, 0.4, 0.4 ],
                     [ 0.5, 0.5, 0.5, 0.5, 0.5, 0.5 ]
                ]
            },
155             {
                "a": [ 0, 1, 2, 3 ],
                "b": [
160                     [ 0.1, 0.1, 0.1, 0.1, 0.1, 0.1 ],
                     [ 0.2, 0.2, 0.2, 0.2, 0.2, 0.2 ],
                     [ 0.3, 0.3, 0.3, 0.3, 0.3, 0.3 ],
                     [ 0.4, 0.4, 0.4, 0.4, 0.4, 0.4 ],
                     [ 0.5, 0.5, 0.5, 0.5, 0.5, 0.5 ]
                ]
            },
165             {
                "a": [ 0, 1, 2, 3 ],
                "b": [
170                     [ 0.1, 0.1, 0.1, 0.1, 0.1, 0.1 ],
                     [ 0.2, 0.2, 0.2, 0.2, 0.2, 0.2 ],
                     [ 0.3, 0.3, 0.3, 0.3, 0.3, 0.3 ],
                     [ 0.4, 0.4, 0.4, 0.4, 0.4, 0.4 ],
                     [ 0.5, 0.5, 0.5, 0.5, 0.5, 0.5 ]
                ]
            }
        ],
175         "layout": "H5D_CONTIGUOUS"
    }
],
"datatypes": [
{

```

```
180     "id": "a93ff089-d466-44e7-b3f0-09db34ec2ef5",
      "type": {
        "class": "H5T_COMPOUND",
        "members": {
185           "a": {
              "class": "H5T_ARRAY",
              "base": "H5T_STD_I32BE",
              "dim": [4]
            },
            "b": {
190               "class": "H5T_ARRAY",
              "base": "H5T_IEEE_F32BE",
              "dim": [ 5, 6 ]
            }
          }
        }
195     }
  ]
}
```

Bibliography

- [BNFDDL] *DDL in BNF for HDF5* [<http://www.hdfgroup.org/HDF5/doc/dcl.html>]. The HDF Group. 2010.
- [DIAL] *Data and Information Access Link* [http://aits.gmu.edu/dial_index.html]. LAITS, George Mason University. 2002.
- [Fielding2000] Roy Thomas Fielding. *Architectural Styles and the Design of Network-based Software Architectures* [<http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>]. Dissertation. University of California, Irvine. 2000.
- [h5py] *HDF5 for Python* [<http://www.h5py.org/>]. Andrew Collette. 2012.
- [HDF5 Infoset] *The HDF5 Information Set*. Mike Folk, Gerd Heber, and Quincey Koziol. The HDF Group. 2013 [To appear].
- [HTTPHandbook] *HTTP*. Developer's Handbook. Chris Shiflett. Sams Publishing. 2003.
- [iRODS] *HDF5-iRODS Project* [<http://www.hdfgroup.org/projects/irods/>]. The HDF Group. 2011.
- [.NET REST] *Effective REST Services via .NET*. For .NET Framework 3.5. Kenn Scribner and Scott Seely. Addison Wesley. 2009.
- [OPeNDAP] *OPeNDAP* [<http://www.opendap.org/>]. OPeNDAP, Inc.. 2012.
- [Pomegranate] *Pomegranate* [<http://pomegranate.nasa.gov/>]. Jet Propulsion Laboratory. 2010.
- [RESTCookbook] *RESTful Web Services Cookbook*. Subbu Allamaraju. O'Reilly. 2010.
- [SDB] *Scientific Data Browser* [<http://hdf.ncsa.uiuc.edu/sdb/sdb.html>]. NCSA. 2000.