*HPC Data MANAGEMENT*

API Specification

Version *1.0*

*08/02/2018*

**Version History**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Version Number** | **Implemented**  **By** | **Revision**  **Date** | **Approved**  **By** | **Approval**  **Date** | **Description of Change** |
| 1.0 | Prasad Konka | 1/25/2016 |  |  | DRAFT |
| 1.0 | Zhengwu Lu | 1/26/2016 |  |  | Review |
| 1.0 | Prasad Konka | 1/26/2016 |  |  | Review updates |
| 1.0 | George Zaki | 2/4/2016 |  |  | Review |
| 1.0 | Prasad Konka | 2/4/2016 |  |  | Review updates |
| 1.0 | Prasad Konka | 2/25/2016 |  |  | Review updates |
| 1.0 | George Zaki | 2/25/2016 |  |  | Review |
| 1.0 | Eran Rosenberg | 4/19/2016 |  |  | S3/Cleversafe Data Registration |
| 1.0 | Prasad Konka | 5/20/2016 |  |  | Service account updates |
| 1.0 | Prasad Konka  Zhengwu Lu | 6/2/2016 |  |  | Updates per Greg Warth comments |
| 1.0 | Prasad Konka | 7/15/2016 |  |  | System generated metadata |
| 1.0 | Prasad Konka | 8/18/2016 |  |  | Notifications, reports |
| 1.0 | Eran Rosenberg | 8/24/2016 |  |  | Get notification subscriptions |
| 1.0 | Eran Rosenberg | 10/23/2016 |  |  | Added documentation for compound metadata query search |
| 1.0 | Eran Rosenberg | 11/2/2016 |  |  | Save Query API |
| 1.0 | Zhengwu Lu | 11/7/2016 |  |  | Revise comment on complex search capability associated with the search APIs. |
| 1.0 | George Zaki and Prasad Konka | 11/21/2016 |  |  | Revised documentation for the compound query |
| 1.0 | Zhengwu Lu | 11/22/2016 |  |  | Review and revise – overall minor (format, typo, and omission) |
| 1.0 | Zhengwu Lu | 12/12/2016 |  |  | Revise search default implementation on level specification, and remove simple search |
| 1.0 | Eran Rosenberg | 12/15/2016 |  |  | Update error messages for search |
| 1.0 | Prasad Konka | 12/19/2016 |  |  | Removed hpc-server from path. Updated WADL schema. |
| 1.0 | Eran Rosenberg | 1/8/2017 |  |  | Added total count todata search services |
| 1.0 | Eran Rosenberg | 2/7/2017 |  |  | Updated notification subscription API |
| 1.0 | Eran Rosenberg | 2/7/2017 |  |  | Updated named query Add/Update/Get API |
| 1.0 | Eran Rosenberg | 3/15/2017 |  |  | Update collection registration w/ parent folders |
| 1.0 | Eran Rosenberg | 3/21/2017 |  |  | Updated API for user, and group API |
| 1.0 | Eran Rosenberg | 3/22/2017 |  |  | Updated Permissions API |
| 1.0 | Eran Rosenberg | 4/4/2017 |  |  | Updated User API |
| 1.0 | Eran Rosenberg | 4/22/2017 |  |  | Update for metadata query w/ matching any attribute |
| 1.0 | Eran Rosenberg | 6/5/2017 |  |  | Added delete data-object section |
| 1.0 | Eran Rosenberg | 6/16/2017 |  |  | Add data-model service |
| 1.0 | Eran Rosenberg | 6/21/2017 |  |  | Add bookmark services |
| 1.0 | Eran Rosenberg | 7/20/2017 |  |  | Data Object and collection download refactored services |
| 1.0 | Eran Rosenberg | 10/14/2017 |  |  | Added support for checksum validation |
| 1.0 | Eran Rosenberg | 10/30/2017 |  |  | Added documentation for Globus directory registration |
| 1.0 | Eran Rosenberg | 11/12/2017 |  |  | Added delete collection spec |
| 1.0 | William Liu | 11/16/2017 |  |  | Added specs for:   1. get permissions on collections 2. get permissions on collections for user |
| 1.0 | William Liu | 11/16/2017 |  |  | Updated service URIs for 2 services:   1. get permissions on collections 2. get permissions on collections for user   Also added additional information about possible error response for the latter of these 2 services. |
| 1.0 | Eran Rosenberg | 12/03/2017 |  |  | Updated data registration w/ API to generate pre-signed upload URL |
| 1.0 | Eran Rosenberg | 1/1/2018 |  |  | Added an option to overwrite files in download requests |
| 1.0 | Eran Rosenberg | 7/7/2018 |  |  | Added documentation for bulk rename API |
| 1.0 | Eran Rosenberg | 7/8/2018 |  |  | Added documentation for rename API of a single collection or Data Object |
| 1.0 | Eran Rosenberg | 8/2/2018 |  |  | Updated rename API to move API |

TABLE OF CONTENTS

[1. Purpose 7](#_Toc521004016)

[2. Introduction 7](#_Toc521004017)

[2.1. What is HPC DME? 7](#_Toc521004018)

[2.2. Intended Users 7](#_Toc521004019)

[2.3. HPC DME URL 8](#_Toc521004020)

[3. Pre-Requisites 8](#_Toc521004021)

[3.1. Share data with HPC DME Service account 8](#_Toc521004022)

[4. HPC DME Overwiew 8](#_Toc521004023)

[4.1 Data Management 8](#_Toc521004024)

[4.2 Business Rules and characteristics of Collections 9](#_Toc521004025)

[4.3 Business Rules and characteristics of Data objects 10](#_Toc521004026)

[4.4 Metadata 10](#_Toc521004027)

[4.5 System generated metadata 16](#_Toc521004028)

[4.6 Metadata HIErarchy 17](#_Toc521004029)

[4.7 Data transfer 18](#_Toc521004030)

[4.8 Notifications 20](#_Toc521004031)

[4.9 Reports 21](#_Toc521004032)

[4.10 Background processes 22](#_Toc521004033)

[5. HPC Server API Specification 22](#_Toc521004034)

[5.1 URL Format 23](#_Toc521004035)

[5.2 Security 23](#_Toc521004036)

[5.3 STATUS CODES 23](#_Toc521004037)

[5.4 Using HPC server API 24](#_Toc521004038)

[5.5 Enroll User 24](#_Toc521004039)

[5.6 Get User 26](#_Toc521004040)

[5.7 Update User 28](#_Toc521004041)

[5.8 Search Active Users 30](#_Toc521004042)

[5.9 Search All Users 31](#_Toc521004043)

[5.10 Create Group 32](#_Toc521004044)

[5.11 Get Group 34](#_Toc521004045)

[5.12 Update Group 35](#_Toc521004046)

[5.13 Search Groups 36](#_Toc521004047)

[5.14 Delete Group 37](#_Toc521004048)

[5.15 Register Collection 38](#_Toc521004049)

[5.16 Find Collection by compound metadata query 44](#_Toc521004050)

[5.17 Find Collection by compound metadata query name 50](#_Toc521004051)

[5.18 Set Collection Permissions 52](#_Toc521004052)

[5.19 Get Collection Permissions 54](#_Toc521004053)

[5.20 GET PERMISSIONS ON COLLECTIONS 56](#_Toc521004054)

[5.21 GET USER PERMISSIONS ON COLLECTIONS 59](#_Toc521004055)

[5.22 Delete Collection 61](#_Toc521004056)

[5.23 Move Collection 63](#_Toc521004057)

[5.24 Register Data File 63](#_Toc521004058)

[5.25 Bulk Data Files Registration 70](#_Toc521004059)

[5.26 Find data by compound metadata query 75](#_Toc521004060)

[5.27 Find data by compound metadata query name 80](#_Toc521004061)

[5.28 Download data file 82](#_Toc521004062)

[5.29 Get data file Download Status 87](#_Toc521004063)

[5.30 Download Collection 88](#_Toc521004064)

[5.31 Get Collection Download Status 90](#_Toc521004065)

[5.32 Download Data Object List 92](#_Toc521004066)

[5.33 Get Data Object List Download Status 93](#_Toc521004067)

[5.34 Delete data file 95](#_Toc521004068)

[5.35 Move data file 97](#_Toc521004069)

[5.36 Set Data Permissions 97](#_Toc521004070)

[5.37 Get Data Permissions 100](#_Toc521004071)

[5.38 Subscribe to notifications 102](#_Toc521004072)

[5.39 Get notification subscription 105](#_Toc521004073)

[5.40 Get User notifications 106](#_Toc521004074)

[5.41 Add user query 108](#_Toc521004075)

[5.42 Update user query 110](#_Toc521004076)

[5.43 Delete user query 112](#_Toc521004077)

[5.44 Get user queries 113](#_Toc521004078)

[5.45 Get user query 115](#_Toc521004079)

[5.46 Get Metadata attributes 118](#_Toc521004080)

[5.47 Get Data Management Model 120](#_Toc521004081)

[5.48 Add Bookmark 130](#_Toc521004082)

[5.49 Update Bookmark 131](#_Toc521004083)

[5.50 Delete Bookmark 133](#_Toc521004084)

[5.51 Get Bookmarks 134](#_Toc521004085)

[5.52 Get Bookmark 135](#_Toc521004086)

[5.53 Generate Report 136](#_Toc521004087)

[5.54 Bulk Move of Collections and Data Objects 139](#_Toc521004088)

[6. APPENDIX A: HPC SERVER API GRAMMER 141](#_Toc521004089)

# Purpose

This HPC API specification is intended to help developers to understand the design and working structure of the HPC data management environment (DME). This specification will explain all the capabilities of this HPC DME and the procedures for data storage, data management, metadata collection, metadata storage and data transferring. This specification steers the users through the process of using the HPC DME for managing large data and enhancing the governance using associated metadata. The specification describes the functional capabilities of this flexibly manageable and multi-functional data archival environment and provides use cases and examples of its functionalities. The specification further details the prerequisites needed for setting up the HPC DME**.**

# Introduction

## What is HPC DME?

The HPC DME, High Performance Computing Data Management Environment, is a highly adaptable and an open-ended data storage environment supporting storage and management of large data, produced from high performance computing systems. HPC DME provides capabilities for storing, managing, transferring and sharing large data across different systems securely and efficiently.

Users can store data on HPC DME for a duration defined by data storage policy, share and transfer their data such that they do not have to redistribute or maintain copies of the data on other systems by eliminating the data integrity issues. HPC DME stores and associates user defined metadata to any registered data at different levels of data life cycle, enabling the DME not only to help identify the data but also enhancing the search capabilities and to be able to attach a value factor to each dataset.

## Intended Users

The HPC data management environment is designed to meet the data storage and data management needs of NCI cancer community. Any user with a valid NCI user account can log into the HPC DME using NCI credentials.

Note: HPC DME currently only supports NCI account holders.

## HPC DME URL

The following is the HPC DME Server API URL:

<https://hpcdmeapi.nci.nih.gov/><Resource Name>

# Pre-Requisites

The following pre-requisite (3.1) is required only if user role requires registering data objects using Globus within the HPC DME.

## Share data with HPC DME Service account

HPC DME uses its internal service account to upload or download data from/to a user’s Globus endpoint to HPC data archive target. In order to perform this action, a user would have to grant read access to upload a data file to the archive and write access to download from the archive to user’s Globus endpoint with HPC service account “ncif-hpcdm-svc”. Please refer to HPC\_User\_Guide on how to best perform such actions.

# HPC DME Overwiew

## Data Management

The HPC data management environment relies on backend object storage systems (Cleversafe object store) to provide a high-reliability storage and an API to manage large data objects life cycle. The data object registration system associates a label with a given managed data file or folder and captures extensible metadata for the managed data object.

The HPC DME is accessed through its application programming interfaces (APIs) to enable users’ scientific data management activities. At a high level, there are two important components in HPC data management environment. 1) Metadata management: HPC DME by default integrates with iRODS iCAT instance to manage metadata and its security for both collections and data objects. 2) Data transfer: HPC DME supports multiple types of data transfers such as Amazon S3 and Globus Connect. It uses Globus to perform asynchronous data transfer between Globus endpoints. It also supports synchronous data transfers from an accessible file system to Cleversafe object store. HPC DM pluggable architecture allows both these implementations to be replaced with alternatives easily while keeping its APIs unchanged. The basic features of HPC DME is to help users in registering and uploading their data to the HPC DME archive storage and managing it. HPC DME archive storage can be a permanent storage for the users’ data and can be used as a platform to search, manage and transfer the data to other storage systems and also to share with other collaborators or users. Each data object is stored along with its required and user defined metadata. The associated metadata can be used as search criteria to identify dataset(s) through HPC DME API.

The HPC data management implementation provides users flexibility to define collection types and metadata attributes. Collections in HPC DME can be Projects, Datasets, Samples and Runs. A data object is an individual file or a set of files in a folder. Collections can be of different types. Each collection can have multiple subcollecitons if necessary. For example, a dataset can be a subcollection of a sequencing project, imaging study or an analysis project. A project can be an entity (umbrella project) indicating a scientific ‘study’.

## Business Rules and characteristics of Collections

* Each collection can be referred with a logical path (“collection\_path”) which is unique across HPC DME.
* Irrespective of the logical path, each collection is associated with UUID metadata attribute to uniquely identify itself.
* A collection can be a type of one of these predefined types – Project, Dataset, Folder. Additional collection types such as Runs, Samples may be custom added and configured in the system through metadata configuration. Please consult your HPC DME system administrator for details.
* Users can create/register multiple collections with HPC DME.
* Each collection has its own required set of metadata which needs to be submitted at the time of its registration. This required metadata can be configured by a system administrator through updating the metadata policy file on the server. Only system admin can update this metadata policy file on the server. There is no UI or API to update the policy file at this time. Please refer to HPC\_Admin\_Guide for more details.
* Users have an option of adding new metadata variables to the required metadata associated with each collection.
* A collection can contain one or more data objects, or child collections. This is achieved using the logical path of a given collection.

## Business Rules and characteristics of Data objects

* A data object can be a single data file or a folder compressed or uncompressed.
* Each data object can be referred with a logical path (“object\_path”) which is unique across HPC DME.
* A parent collection should be registered first with HPC DME before registering a data object under the collection.
* Irrespective of the logical path, each data object is associated with UUID metadata attribute to uniquely identify itself.
* Each data object has associated required metadata which is required to be submitted at the time of data object registration.
* Users have an option of adding new metadata variables to the required metadata associated with each data object.

## Metadata

Metadata is defined as the data about the data. It is the information which describes the actual data such as the date and origin of creation, its contents, its condition, processing it has gone through and associations to other objects etc. Metadata is employed to make data searches faster, more specific and also enable and promote data sharing among scientists.

HPC DME collects metadata for each collection and data object, registered and stored in a database along with the associations. HPC DME collects two kinds of metadata related to a collection or a data object, namely, administrative and center/division specific. The administrative metadata is the required set of information which needs to be submitted at the time of registration with HPC DME. New metadata variables can be added to both administrative and center/division specific metadata sets after a user obtains proper authorizations and permissions. The metadata can also be updated by authorized users.

One of the primary functions of HPC DME is to connect unstructured data with metadata. Metadata may be attached to files, folders and collections. HPC DME stores metadata in the form of “triples” to its relational database. The triples consist of an attribute field, a value field, and a unit field. The content of each of these fields can be independently defined and applied. This Metadata can be changed and updated through the life cycle of each data object.

The following table shows default required Metadata for a dataset in the HPC DME. This list can be modified by a system administrator to add, update or remove any of these attributes. Details on updating metadata attributes configuration is detailed in the section below.

**Metadata for collection:**

The following table shows default required Metadata for a collection (such as a project, study, or dataset) in the HPC DME.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Metadata Variable** | **Definition** | **Default** |
| **Administrative Metadata** | | | |
|  | collection\_type | Collection type name (Default valid values are Project, Dataset, Folder) |  |
|  | name | Name for the collection as provided by the depositor |  |
|  | description | Description of collection |  |
|  | source\_lab\_pi | PI of the lab of the depositor at the time of deposit |  |
|  | lab\_branch | Lab or Branch or Program the PI belongs to |  |
|  | pi\_doc | Division, Organization, Center the PI belongs to |  |
|  | original\_date\_created | Date the collection was created originally | Date the dataset was deposited |
|  | data\_creator | Person or Organization lead who created the collection | Not Specified |
|  | phi\_content | Presence of Protected Health Information in the collection (such as dataset) deposited via HPC DME.  Valid values are (PHI Present, PHI Not Present, Not Specified) | Not Specified |
|  | pii\_content | Presence of Personally Identifiable Information in the collection (such as dataset) deposited via HPC DME. Valid values are (PII Present, PII Not Present, Not Specified) | Not Specified |
|  | data\_encryption\_status | If the data is encrypted or not. Valid values are (Encrypted,  Not Encrypted, Not Specified) | Not Specified |
|  | data\_compression\_status | If the data is compressed or not | Not Specified |
|  | funding\_organization | Organization Funding the generation of Data | Not Specified |
| **Division/Center specific metadata** | | | |
|  | comments | General text for internal use and reference |  |

**System Generated Metadata Variables for any Collection:**

|  |  |  |
| --- | --- | --- |
| Id | local identifier (serves as foreign key to connect to other metadata in data management system |  |
| uuid | universal dataset identifier (reserved for future use: default is ‘unspecified’) |  |
| create\_date | Date the collection (project, dataset etc) was registered with HPC DME. |  |
| update\_date | Date the collection (project, dataset etc) was updated with HPC DME. |  |
| registerd\_by | User depositing the collection |  |
| registered\_by\_name | Name of the person registering the collection |  |
| configuration\_id | The data management configuration ID |  |

The following table shows default required Metadata for a data object in the HPC DME.

**Metadata for Data object/file:**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Metadata Variable** | **Definition** | **Default** |
| **Administrative metadata** | | | |
|  | name | Name for the file as provided by the depositor |  |
|  | description | Extensible description of File |  |
|  | source\_lab\_pi | PI of the lab of the depositor at the time of deposit | Auto Populated |
|  | lab\_branch | Lab or Branch or Program the PI belongs to |  |
|  | pi\_doc | Division, Organization, Center the PI belongs to |  |
|  | original\_date\_created | Date the File was created originally | Date the file was deposited |
|  | data\_creator | Person or Organization lead who created the data | Not Specified |
|  | phi\_content | Presence of Protected Health Information in the datasets deposited via HPC DME.  Valid values are (PHI Present, PHI Not Present, Not Specified) | Not Specified |
|  | pii\_content | Presence of Personally Identifiable Information in the datasets deposited via HPC DME. Valid values are (PII Present, PII Not Present, Not Specified) | Not Specified |
|  | data\_encryption\_status | If the data is encrypted or not. Valid values are (Encrypted,  Not Encrypted, Not Specified) | Not Specified |
|  | data\_compression\_status | If the file is compressed or not | Not Specified |
|  | funding\_organization | Organization Funding the generation of Data | Not Specified |
| **Division/Center specific metadata** | | | |
|  | comments | General text for internal use and reference |  |
| **System Generated Metadata Variables for any Data object / file (Automated)** | | | |
|  | id | local identifier (serves as foreign key to connect to other metadata in data management system |  |
|  | uuid | universal dataset identifier (reserved for future use: default is ‘unspecified’) |  |
|  | dataset\_signature\_md5 | MD5 checksum |  |
|  | Dataset\_signature\_sha1 | SHA1 checksum |  |
|  | Data\_size | Size of the dataset being deposited |  |
|  | create\_date | Date the data file was registered with HPC DME. |  |
|  | update\_date | Date the data file was updated with HPC DME. |  |
|  | registerd\_by | User depositing the data file |  |
|  | registered\_by\_name | Name of the person registering the data file |  |
|  | Configuration\_id | The data management configuration ID |  |
|  | source\_file\_container\_id | The container ID of the data source. | Globus Only |
|  | source\_file\_id | The file id of the data source. | Globus Only |
|  | archive\_file\_container\_id | The container ID of the data object in the Archive |  |
|  | archive\_file\_id | The file ID of the data object in the Archive |  |
|  | data\_transfer\_request\_id | The data transfer request ID | Globus Only |
|  | data\_transfer\_status | The status of the data transfer |  |
|  | data\_transfer\_type | The method used to transfer the data |  |
|  | Source\_file\_size | The file size of the data object source | Globus Only |

HPC DME provides a flexible way to configure metadata attributes through a policy file. The following is an example of the policy file.

|  |
| --- |
| **Sample Policy File** |
| {"HpcMetadataValidationRules": {  "collectionMetadataValidationRules": [  {  "attribute": "Collection type",  "mandatory": true,  "validValues": [  "project",  "dataset",  "folder"  ],  "ruleEnabled": true,  "DOC": "DOC-NAME"  },  {  "attribute": "Project name",  "mandatory": true,  "collectionType": "project",  "ruleEnabled": true,  "DOC": "DOC-NAME"  },  {  "attribute": "Project type",  "mandatory": true,  "defaultValue": "Unspecified",  "collectionType": "project",  "validValues": [  "Umbrella Project",  "Sequencing",  "Analysis"  ],  "ruleEnabled": true,  "DOC": "DOC-NAME"  }, |

* attribute: Metadata attribute name
* mandatory: Flag to indicate if it is required or not. Valid values are “true”, “false”
* defaultValue: Default value of the attribute if no value is given
* collectionType: Collection type name applicable for this attribute rule. Only one value is allowed.
* validValues: List of valid values for this attribute
* ruleEnabled: Flag to indicate if this rule is enabled or not. Valid values are “true” or “false”. If value is set to “false”, this rule will not be evaluated during validation process.
* DOC: Division name applicable. Any registered user account is assigned one DOC in the system. Any specific DOC dependent rules may be attached and activated.

## System generated metadata

Apart from user provided metadata, HPC DME generates system metadata to track information for security and reporting purposes. This read only metadata can be viewed by authorized users. Metadata policy file mentioned in the section above, has configuration information for system metadata. The following are the system metadata attributes supported at this time. System Administrator may remove some of these attributes from the list as needed. Removing an attribute from the policy would result not recording that attribute as part of collection or data object metadata.

**System Metadata for Collections**

|  |  |
| --- | --- |
| Attribute Name | Description |
| uuid | Collection unique identifier |
| registered\_by | Collection registered by UserId |
| registered\_by\_name | Collection registered by user name |
| configuration\_id | Collection registered using this data management configuration |

**System Metadata for Data objects**

|  |  |
| --- | --- |
| Attribute Name | Description |
| uuid | Collection unique identifier |
| registered\_by | Collection registered by UserId |
| registered\_by\_name | Collection registered by user name |
| configuration\_id | Collection registered w/ this data management configuration |
| source\_file\_container\_id | Source data object container identifier (Ex: Globus endpoint Id) |
| source\_file\_id | Source data object path |
| archive\_file\_container\_id | Archive data object container identifier |
| archive\_file\_id | Archive data object path |
| data\_transfer\_request\_id | Data transfter request identifier |
| data\_transfer\_status | Data transfer status |
| data\_transfer\_type | Data transfer type |
| source\_file\_size | Source file size |
| archive\_caller\_object\_id | archive\_caller\_object\_id is optional system metadata. It is populated when "callerObjectId" is set as part of object registration request. "callerObjectId" string is additional path information to calculate destination path. By default it is "/", so user do not have to provide it. |

## Metadata HIErarchy

Metadata can be added to a collection or a data file. These collections or data files can be logically structured in a hierarchical manner. A hierarchy is typically viewed as a pyramid and can be depicted with a tree structure. In order to search for a collection or a data file by its metadata or its parent metadata, HPC DME custom solution, by default, inherits parent metadata to its children. Child entities can always override inherited parent metadata attributes. This approach facilitates advanced and efficient search capabilities where you can find data files based on its own or its parent metadata attributes. For example, you can find all data files (child) for a project (parent) with name “xyz”.

HPC DME also supports limiting your search to a hierarchy level. For example, if your data hierarchy is logically structures as /PI\_lab/Project/Run/Sample/Datafile:

* All PI\_lab metadata is inherited to Project and the level of this hierarchy is 5
* All Project metadata is inherited to Run and the level of this hierarchy is 4
* All Run metadata is inherited to Sample and the level of this hierarchy is 3
* All Sample metadata is inherited to Datafile and the level of this hierarchy is 2
* The level of Datafile hierarchy is 1.

When you are searching for an entity (collection or data file), you could limit your search to a level using comparison operators like EQUAL, NOT\_EQUAL, LIKE, NUM\_LESS\_THAN, NUM\_LESS\_OR\_EQUAL, NUM\_GREATER\_OR\_EQUAL, NUM\_GREATER\_THAN.

Using ‘level’ and ‘levelOperator’ as part of your search is optional. If not provided, the search will find the matched data files per the metadata at all levels (collections and data files) by default. The rationale for this is to enable broader search support in cases where users may foeget the exact levels certain metadata attributes may be associated with. Similarly, search will match metadata found at any level of collections if the search is indicated for collections specifically. The rationale for the approach is that there is nothing harmful to show a broader collections result set regardless if any data objects have been registered under certain collection path.

## Data transfer

HPC DME is configured to work with both synchronous and asynchronous data transfers. HPC DME is configured to use Cleversafe object store as its data archival storage. Data from a local or shared location can be transferred to HPC data archive system synchronously. For large data object data transfer, it is recommended to host your data at a Globus endpoint so that the transfer can be asynchronous. Please see <https://www.globus.org/globus-connect-server> for more details. HPC DM Server API integrates with Globus Connect API to provide seamless data transfer, tracking and reporting capabilities. Since there is no Globus connect plugin to work with Cleversafe directly, HPC DME provides 2-hop solution to support data transfer from/to a Globus endpoint to/from HPC DME Cleversafe data store. As depicted in the diagram below, to enable 2-hop transfer, HPC DME uses temporary storage to stage data while transferring. Stage data gets cleaned as soon as the data transfer is complete.



Upon dataset registration, HPC DME initiates data transfer with Globus Connect Server asynchronously and keep the information within HPC DM database for reporting purposes. If the data transfer is done with 2-hop approach, HPC DME uses the following status codes to accurately represent data transfer status.

**IN\_PROGRESS\_TO\_TEMPORARY\_ARCHIVE**: This is the status code while data is transferred from source location to HPC DME temporary storage.

**IN\_TEMPORARY\_ARCHIVE:** This is the status code if the data is transferred to HPC DME temporary storage and it is waiting to be transferred to archive storage by HPC DME.

**IN\_PROGRESS\_TO\_ARCHIVE:** This is the status code while data is transferred from HPC DME temporary storage to HPC DME archive storage.

**ARCHIVED**: This is the status code if the data is transferred to HPC DME archive storage.

**FAILED**: This is the status code if the data transferring failed for any known reasons.

**UNKNOWN**: This is the status code if the data transferring failed for any unknown reasons.

HPC DME background process checks for any “IN\_TEMPORARY\_ARCHIVE” objects and initate data transfer to archive storage. By default, this background process runs every 60 seconds and it is configurable. Background process will update data transfer status with HPC DME and clean up data from stage area.

Additionally, HPC DME supports synchronous data transfer from a source with no Globus endpoint. As depicted in the diagram below, HPC DME client initiate synchronous data registration request where data is transferred to the archive storage via HPC DME server.



## Notifications

HPC DME generates different events while processing data upload, download and sharing requests. Authorized users can subscribe to these event notifications via email or text. Whenever these events happen, HPC background process pushes out notifications to subscribed users.

The following event types can be subscribed by users:

**DATA\_TRANSFER\_UPLOAD\_IN\_TEMPORARY\_ARCHIVE:**

This event is generated when uploaded data is moved to temporary storage before moving it into archive storage. HPC DME uses 2-hop upload when using asynchronous data transfer from a Globus endpoint. As 1st hop, HPC DME transfers data from source Globus endpoint to HPC DME stage globus endpoint. Once the transfer is successful, 2nd hop of tranfering data to archive storage is initiated.

**DATA\_TRANSFER\_UPLOAD\_ARCHIVED:**

This event is generated when uploading data to the archive storage request is completed.

**DATA\_TRANSFER\_UPLOAD\_FAILED:**

This event is generated when uploading data to the archive storage request is failed.

**DATA\_TRANSFER\_DOWNLOAD\_COMPLETED:**

This event is generated when downloading data from the archive storage request is completed.

**DATA\_TRANSFER\_DOWNLOAD\_FAILED:**

This event is generated when downloading data from the archive storage request is failed.

**USAGE\_SUMMARY\_REPORT:**

This event is generated when the background process generates data archive system usage summary report.

**USAGE\_SUMMARY\_BY\_WEEKLY\_REPORT:**

This event is generated when the background process generates data archive system weekly usage summary report.

**COLLECTION\_UPDATED:**

This event is generated when a collection is updated. An update can be metadata update, sub-collection registration or data object registration to the collection of interest.

## Reports

Users of HPC DME can generate different summary reports useful to analyze its usage. Default HPC DME report provides the following information:

|  |
| --- |
| Total registered users:  Total size of the data: bytes  Largest file: bytes  Average file size: bytes  Total number of objects:  Total number of collections:  Project:  Dataset:  ..  Total number of metadata attributes:  Total number of files with size:  < 1MB:  > 1MB and < 10MB:  > 10MB and < 50MB:  > 50MB and < 100MB:  > 100MB and < 500MB:  > 500MB and < 1GB:  > 1GB and < 10GB:  > 10GB: |

HPC DME provides REST API to generate the following reports.

**USAGE\_SUMMARY\_REPORT:**

This report provides summarized report from inception to till date.

**USAGE\_SUMMARY\_BY\_WEEKLY\_REPORT:**

This report provides summarized report for past 1 week from today.

**USAGE\_SUMMARY\_BY\_DOC\_REPORT:**

This report provides summarized report for a given DOC(S) from inception to till date.

**USAGE\_SUMMARY\_BY\_DOC\_BY\_WEEKLY\_REPORT:**

This report provides summarized report for a given DOC(S) for past 1 week from today.

**USAGE\_SUMMARY\_BY\_USER\_REPORT:**

This report provides summarized report for a given user(s) from inception to till date.

**USAGE\_SUMMARY\_BY\_USER\_BY\_WEEKLY\_REPORT:**

This report provides summarized report for a given user(s) for past 1 week from today.

## Background processes

HPC DME Server runs background processes to synchronize some of its action with user actions. Timing of these background processes can be configured through hpc-server-\*.properties file. It is recommended not to change these settings unless there is a need. The default timings are set to avoid any race conditions with system resources. The following are the properties from hpc-server-\*.properties file.

# Runs daily at 12AM

#hpc.scheduler.cron.summaryreport.delay=0 0 23 1/1 \* ?

# Runs Weekly at 12AM

#hpc.scheduler.cron.weeklysummaryreport.delay=0 23 \* \* 7 ?

#Runs every minute at 10 secs

#hpc.scheduler.cron.processevents.delay=10 0/1 \* \* \* ?

#Runs every minute at 15 secs

#hpc.scheduler.cron.updateDataTransferUploadStatus.delay=15 0/1 \* \* \* ?

#Runs every minute at 25 secs

#hpc.scheduler.cron.processTemporaryArchive.delay=25 0/1 \* \* \* ?

#Runs every minute at 35 secs

#hpc.scheduler.cron.cleanupDataTransferDownloadFiles.delay=35 0/1 \* \* \* ?

#Runs every 15 minutes

#hpc.scheduler.cron.refreshMaterializedViews.delay=0 0/15 \* \* \* ?

# HPC Server API Specification

Server API can be categorized into the following sections. HPC DME Server API specifications will be revised as more functionality or stabilization is added.

* User
  + [Enroll user into HPC DME](#_Enroll_User) (Create)
  + Get User
  + Update an existing User
* Collection
  + [Register Collection with its metadata into HPC DME](#_Register_Project)
  + Find Collection by path
  + [Find Collection by metadata](#_Find_Project_by_1)
* Data file
  + [Register data file along with metadata into HPC DME](#_Register_Dataset) (Create and Update)
  + Find Data Object by path

[Find Data Object by metadata](#_Find_Dataset_by_2)

* Permissions
  + Assign permissions on iRODS resources
* Download
  + Download data object/file
* Notifications
  + Subscribe to notifications
* Reports
  + Generate reports

## URL Format

HPC Server API URL is always relative to the following HPC Server base URL.

<http://hpcdmeapi.nci.nih.gov>

## Security

Each HPC Server API call is secured by authentication and authorization. HPC Server API is configured to support BASIC authentication credentials over HTTPS. Given credentials are verified against NCI LDAP for authentication. An authenticated User is verified with HPC security system for authorization.

Users of HPC API interface are required to enroll into HPC DME to start with. Please consult HPC DME administrator for an account at [HPC\_DME\_Admin@nih.gov](mailto:HPC_DME_Admin@nih.gov) after obtaining your group or DOC approval of using the HPC DME.

HPC Server API internally interacts with iRODS Jargon API to work with iRODS securely. iRODS Jargon APIs are protected with PAM *(Pluggable Authentication Modules)* authentication via NCI LDAP credentials.

## STATUS CODES

HPC API returns the following error object in case of any exceptions.

*{"gov.nih.nci.hpc.dto.error.HpcExceptionDTO": {*

*"errorType": "HPC Error Type",*

*"message": "Error Message",*

*"stackTrace": "Error Stacktrace”*

*}}*

The following are HPC Error Types:

SPRING\_CONFIGURATION\_ERROR,

INVALID\_REQUEST\_INPUT,

UNAUTHORIZED\_REQUEST,

REQUEST\_AUTHENTICATION\_FAILED,

REQUEST\_REJECTED,

DATABASE\_ERROR,

DATA\_TRANSFER\_ERROR,

UNEXPECTED\_ERROR;

Each of HPC Error codes are mapped the following HTTP codes.

|  |  |  |
| --- | --- | --- |
| HTTP Code | Status | Description |
| 200 | OK | Query request is processed successfully. Query result may include partially successful results in case of batch processing. |
| 201 | Created | Created requested resource |
| 400 | Bad Request | Request could not be understood by the HPC server due to malformed syntax. |
| 401 | Unauthorized | User authentication is required. The request either did not provide correct credentials or the user does not have permission. |
| 500 | Internal Server Error | Internal error was encountered on the HPC server. This will be a non-recoverable error. Details are provided in message body. |

## Using HPC server API

HPC Server API follows REST specification. Any client program or language supporting REST specification can be used to interact with HPC API. Each of the API specifications given below has example Java client code via accessing HPC REST API interfaces. Client interface like SOAPUI (<http://www.soapui.org/>) can also be used to test HPC Server APIs. Installing SOAPUI instructions can be found at <http://www.soapui.org/getting-started/installing-soapui/installing-on-windows.html>

## Enroll User

|  |  |
| --- | --- |
| Title | Enroll User |
| Description | Enroll user into HPC system. In order to use HPC DME, a user must be enrolled into the DME. Only users with “SYSTEM\_ADMIN”, “GROUP\_ADMIN” roles are authorized to run this API.  “userRole” attribute is optional with the request. If not given, user is enrolled as “USER” role. Valid values are “USER”, “SYSTEM\_ADMIN” and “GROUP\_ADMIN”. However, there is a known issue with iRODS API to create “GROUP\_ADMIN” role. Work-around is to first register users with “USER” role, then update “USER” role to “GROUP\_ADMIN” through iCommand interface. A group admin can create a user for his/her own DOC only.  By default, an iRODS account will be created for the enrolled user, unless an account is already created |
| URL | /user/{nciUserId} |
| Method | PUT |
| Acceptable request representation | application/json  application/xml |
| Available response representation |  |
| URL Params | nciUserId – the NCI User ID to enroll |
| Data Params | **JSON:**  {  "firstName": "John",  "lastName": "Doe",  “defaultBasePath” : “/FNLCR”,  "userRole": “USER”  } |
| Success Response | HTTP/1.1 201 Created  Location: https://hpcdmeapi.nci.nih.gov/user/u1c06009i |
| Error Response | **UserId already exists:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  {"gov.nih.nci.hpc.dto.error.HpcExceptionDTO":{"errorType":"REQUEST\_REJECTED","requestRejectReason":"USER\_ALREADY\_EXISTS","message":"User already exists: nihUserId = u1c06009i","stackTrace":"gov.nih.nci.hpc.exception.HpcException: User already exists: r\n"}}  **Invalid User DOC:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  {  "errorType": "INVALID\_REQUEST\_INPUT",  "message": "Invalid DOC: FNLCR1. Valid values: [FNLCR, CCBR]",  "stackTrace": "gov.nih.nci.hpc.exception.HpcException: Invalid DOC: FNLCR1. Valid values: [FNLCR, CCBR][INVALID\_REQUEST\_INPUT] "  }  **Invalid User Role:**  {  "errorType": "INVALID\_REQUEST\_INPUT",  "message": "Invalid user role: USER1. Valid values: [USER, GROUP\_ADMIN, SYSTEM\_ADMIN, NOT\_REGISTERED]",  "stackTrace": "gov.nih.nci.hpc.exception.HpcException: Invalid user role: USER1. Valid values: [USER, GROUP\_ADMIN, SYSTEM\_ADMIN, NOT\_REGISTERED][INVALID\_REQUEST\_INPUT]\ "  }  **Not Authorized:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  {  "errorType": "REQUEST\_REJECTED",  "requestRejectReason": "NOT\_AUTHORIZED",  "message": "Not authorizated to update frist name, last name, DOC, role. Please contact system administrator",  "stackTrace": "gov.nih.nci.hpc.exception.HpcException: Not authorizated to update frist name, last name, DOC, role. Please contact system administrator[REQUEST\_REJECTED]"  } |

## Get User

|  |  |
| --- | --- |
| Title | Get User |
| Description | Get user name, doc and role based on user-id |
| URL | /user/{nciUserId} |
| Method | GET |
| Acceptable request representation | application/json  application/xml |
| Available response representation |  |
| URL Params | nciUserId – the NCI user ID. This is optional - if not provided, the invoker user details are returned. |
| Data Params |  |
| Success Response | HTTP/1.1 200 OK  {  "firstName": "John",  "lastName": "Doe",  “defaultBasePath” : “/FNLCR”,  "userRole": “USER”,  “active” : true  } |
| Error Response | **Null NCI User ID:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  {"gov.nih.nci.hpc.dto.error.HpcExceptionDTO":{"errorType":"REQUEST\_REJECTED","requestRejectReason":"INVALID\_REQUEST\_INPUT","message":"Null NCI User ID","stackTrace":"gov.nih.nci.hpc.exception.HpcException: Null NCI User ID: r\n"}}  **Unauthorized access request:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  {  "errorType": "UNAUTHORIZED\_REQUEST",  "message": "Unauthorized access request",  "stackTrace": "gov.nih.nci.hpc.exception.HpcException: Unauthorized access request "  }  **Failed to get a user:**  {  "errorType": "DATABASE\_ERROR",  "message": "Failed to get a user: ",  "stackTrace": "gov.nih.nci.hpc.exception.HpcException: Failed to get a user\ "  } |

## Update User

|  |  |
| --- | --- |
| Title | Update User |
| Description | Update an existing user in the HPC system.   * Only SYSTEM\_ADMIN can update any user. * The update request reports an error if it contains information that is not authorized to be updated (e.g a self update request from a user except system\_admin). * System\_admin user type is not allowed to downgrade self role.   A user can be deactivated by setting the ‘active’ indicator to false, or re-activated by setting the indicator to true. When user is inactive, he/she will not have access to the system and the service to search users will not include him/her in search results |
| URL | /user/{nciUserId} |
| Method | POST |
| Acceptable request representation | application/json  application/xml |
| Available response representation |  |
| URL Params | nciUserId – the NCI user ID to be updated |
| Data Params | **JSON:**  {  "firstName": "John",  "lastName": "Doe",  “defaultBasePath” : “/FNLCR”,  "userRole": “USER”,  “active” : true  } |
| Success Response | HTTP/1.1 200 OK |
| Error Response | **UserId does not exists:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  {  "errorType": "REQUEST\_REJECTED",  "requestRejectReason": "INVALID\_NCI\_ACCOUNT",  "message": "User does not exists: nciUserId = batchAdmin16",  "stackTrace": "gov.nih.nci.hpc.exception.HpcException: User does not exists: nciUserId = batchAdmin16 "  }  **Invalid User Input:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  {  "errorType": "INVALID\_REQUEST\_INPUT",  "message": "Invalid add user input",  "stackTrace": "gov.nih.nci.hpc.exception.HpcException: Invalid update user input "  }  **Invalid User DOC:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  {  "errorType": "INVALID\_REQUEST\_INPUT",  "message": "Invalid DOC: FNLCR1. Valid values: [FNLCR, CCBR]",  "stackTrace": "gov.nih.nci.hpc.exception.HpcException: Invalid DOC: FNLCR1. Valid values: [FNLCR, CCBR][INVALID\_REQUEST\_INPUT] "  }  **Invalid User Role:**  {  "errorType": "INVALID\_REQUEST\_INPUT",  "message": "Invalid user role: USER1. Valid values: [USER, GROUP\_ADMIN, SYSTEM\_ADMIN, NOT\_REGISTERED]",  "stackTrace": "gov.nih.nci.hpc.exception.HpcException: Invalid user role: USER1. Valid values: [USER, GROUP\_ADMIN, SYSTEM\_ADMIN, NOT\_REGISTERED][INVALID\_REQUEST\_INPUT]\ "  }  **Not Authorized:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  {  "errorType": "REQUEST\_REJECTED",  "requestRejectReason": "NOT\_AUTHORIZED",  "message": "Not authorizated to update frist name, last name, DOC, role. Please contact system administrator",  "stackTrace": "gov.nih.nci.hpc.exception.HpcException: Not authorizated to update frist name, last name, DOC, role. Please contact system administrator[REQUEST\_REJECTED]"  }  **Not Authorized:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  {  "errorType": "REQUEST\_REJECTED",  "requestRejectReason": "NOT\_AUTHORIZED",  "message": "Not authorizated to downgrade self. Please contact system administrator",  "stackTrace": "gov.nih.nci.hpc.exception.HpcException: Not authorizated to downgrade self. Please contact system administrator[REQUEST\_REJECTED]"  } |

## Search Active Users

|  |  |
| --- | --- |
| Title | Search Users |
| Description | Search active users by search criteria – user Id, first name and last name.  If no search criteria are specified then ALL active users are returned. |
| URL | /user/active |
| Method | GET |
| Acceptable request representation | application/json  application/xml |
| Available response representation |  |
| URL Query Params | nciUserId – the NCI user ID search criteria  firstNamePattern – The first name search pattern.  lastNamePattern – The last name search pattern. |
| Data Params |  |
| Success Response | HTTP/1.1 200 OK  {"users": [        {        "userId": "abc",        "firstName": "first",        "lastName": "last",        “defaultBasePath” : “/FNLCR”,  ]} |
| Error Response | **Null NCI User ID:**  **Failed to get a user:**  {  "errorType": "DATABASE\_ERROR",  "message": "Failed to get a user: ",  "stackTrace": "gov.nih.nci.hpc.exception.HpcException: Failed to get a user\ "  } |

## Search All Users

|  |  |
| --- | --- |
| Title | Search All Users |
| Description | Search all users by search criteria – user Id, first name and last name.  If no search criteria are specified then ALL users (active and inactive) are returned.  Please note that only system admin roles is authorized to call this service. Other users will receive authorization exception. |
| URL | /user/all |
| Method | GET |
| Acceptable request representation | application/json  application/xml |
| Available response representation |  |
| URL Query Params | nciUserId – the NCI user ID search criteria  firstNamePattern – The first name search pattern.  lastNamePattern – The last name search pattern. |
| Data Params |  |
| Success Response | HTTP/1.1 200 OK  {"users": [        {        "userId": "abc",        "firstName": "first",        "lastName": "last",        “defaultBasePath” : “/FNLCR”,  ]} |
| Error Response | **Null NCI User ID:**  **Failed to get a user:**  {  "errorType": "DATABASE\_ERROR",  "message": "Failed to get a user: ",  "stackTrace": "gov.nih.nci.hpc.exception.HpcException: Failed to get a user\ "  } |

## Create Group

|  |  |
| --- | --- |
| Title | Create group and optionally add users to it |
| Description | Create a group and optionally add users to it. |
| URL | /group/{groupName} |
| Method | PUT |
| Acceptable request representation | application/json  application/xml |
| Available response representation | application/json  application/xml |
| URL Params | groupName – the group name to create |
| Data Params | **JSON:**  {  "addUserIds": ["konkapv","aa"]  } |
|  | HTTP/1.1 200 OK  {"addGroupMemberResponses": [        {        "userId": "aa",        "result": false,        "message": "Failed to add group member. Invalid user: aa"     },        {        "userId": "konkapv",        "result": true     }]} |
| Error Response | **Invalid Request:**  The following error is thrown if Group name already exists  HTTP/1.1 400 Bad Request  Content-Type: application/json  {     "errorType": "REQUEST\_REJECTED",     "requestRejectReason": "GROUP\_ALREADY\_EXISTS",  "message": "Group already exists: group-name"  }  **Invalid Group Name:**  The following error is thrown if the group name is invalid. Only alphanumeric, ‘\_’ and ‘-‘ characters are supported  HTTP/1.1 400 Bad Request  Content-Type: application/json  {     "errorType": "INVALID\_REQUEST\_INPUT",  "message": "Invalid group name. Only alphanumeric, '\_' and '-' are supported."  }  **Not Authorized:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  {  "errorType": "REQUEST\_AUTHENTICATION\_FAILED",  "message": "Access Denied: Unauthorized",  "stackTrace": "org.apache.cxf.interceptor.security.AccessDeniedException: Unauthorized "  } |

## Get Group

|  |  |
| --- | --- |
| Title | Get Group |
| Description | Get a group and its users |
| URL | /group/{groupName} |
| Method | GET |
| Acceptable request representation | application/json  application/xml |
| Available response representation |  |
| URL Params | groupName – the group name |
| Data Params |  |
| Success Response | HTTP/1.1 200 OK  {"userIds": [     "konkapv",     "zakigf"  ]} |
| Error Response | **Null Group Name:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  {"gov.nih.nci.hpc.dto.error.HpcExceptionDTO":{"errorType":"REQUEST\_REJECTED","requestRejectReason":"INVALID\_REQUEST\_INPUT","message":"Null NCI User ID","stackTrace":"gov.nih.nci.hpc.exception.HpcException: Null Group Name: r\n"}}  **Unauthorized access request:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  {  "errorType": "UNAUTHORIZED\_REQUEST",  "message": "Unauthorized access request",  "stackTrace": "gov.nih.nci.hpc.exception.HpcException: Unauthorized access request "  } |

## Update Group

|  |  |
| --- | --- |
| Title | Update a group |
| Description | Add / Delete users from an existing group |
| URL | /group/{groupName} |
| Method | POST |
| Acceptable request representation | application/json  application/xml |
| Available response representation | application/json  application/xml |
| URL Params | groupName – the group name to update |
| Data Params | **JSON:**  {  "addUserIds": ["konkapv"],  “deleteUserIds” : [“zakifg”]  } |
|  | HTTP/1.1 200 OK  {     "addGroupMemberResponses":   [ {        "userId": "konkapv",        "result": true     }],     "deleteGroupMemberResponses":   [ {        "userId": "zakifg",        "result": false,        "message": "Failed to delete group member. Invalid user: zakifg"     }]  } |
| Error Response | **Invalid Request:**  The following error is thrown if Group name doesn’t exist  HTTP/1.1 400 Bad Request  Content-Type: application/json  {     "errorType": "INVALID\_REQUEST\_INPUT",  "message": "Group doesn't exist",  }  **Not Authorized:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  {  "errorType": "REQUEST\_AUTHENTICATION\_FAILED",  "message": "Access Denied: Unauthorized",  "stackTrace": "org.apache.cxf.interceptor.security.AccessDeniedException: Unauthorized "  } |
|  |  |

## Search Groups

|  |  |
| --- | --- |
| Title | Search Groups |
| Description | Search groups by a search criteria |
| URL | /group |
| Method | GET |
| Acceptable request representation | application/json  application/xml |
| Available response representation |  |
| URL Params | groupPattern – the search criteria in the form of a SQL ‘LIKE’ matching pattern. If search criteria is omitted than all groups are returned |
| Data Params |  |
| Success Response | HTTP/1.1 200 OK  {"groups": [        {        "groupName": "eran-group",        "userIds":       [           "konkapv",           "rosenbergea"        ]     }  ]} |
| Error Response |  |

## Delete Group

|  |  |
| --- | --- |
| Title | Delete Group |
| Description | Delete an existing group from the HPC system. |
| URL | /group/{groupName} |
| Method | DELETE |
| Acceptable request representation | application/json  application/xml |
| Available response representation |  |
| URL Params | groupName – the group to delete |
| Data Params |  |
| Success Response | HTTP/1.1 200 OK |
| Error Response | **Group does not exists:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  {  "errorType": "REQUEST\_REJECTED",  "requestRejectReason": "INVALID\_NCI\_ACCOUNT",  "message": "Group does not exists: group-name  } |

## Register Collection

Collection registration is a required process. Collections support logical hierarchical structure where a collection can have a parent and many children. Each of these colletions can represent a Project, Dataset or a folder, or any customized category of collection type. More collection types and their metadata are configurable. Based on metadata configuration, a user is mandated to provide required metadata while registering a collection. Similarly, a user needs to register their collections first prior to depositing any data files under these collections.

By default, the service expects the containing collection for the registered collection to exist at time of registration. However, the user can optionally request the parent collection(s) to be created. To do so the ‘createParentCollections’ indicator needs to be set to true, and a list of metadata to create the parent collections needs to be provided. In this case the entire collection hierarchy above the registered collection will be created.

|  |  |
| --- | --- |
| Title | Register Collection |
| Description | A collection registered with HPC DME  By default, the service expects the containing collection for the registered collection to exist at time of registration. However, the user can optionally request the parent collection(s) to be created. To do so the ‘createParentCollections’ indicator needs to be set to true, and a list of metadata to create the parent collections needs to be provided. In this case the entire collection hierarchy above the registered collection will be created. |
| URL | /collection/{path} |
| Method | POST |
| Acceptable request representation | application/json  application/xml |
| Available response representation |  |
| URL Params | {path} – Logical path of the collection to identify with. |
| Data Params | **JSON:**  {  "metadataEntries": [  {  "attribute": "name",  "value": "Demo Project Name10"  },  {  "attribute": "collection\_type",  "value": "Project"  },  {  "attribute": "description",  "value": "Project desc"  },  {  "attribute": "internal\_project\_id",  "value": "ipi"  },  {  "attribute": "source\_lab\_pi",  "value": "Source Lab PI"  },  {  "attribute": "lab\_branch",  "value": "Lab / Branch Name"  },  {  "attribute": "pi\_doc",  "value": "FNLCR"  },  {  "attribute": "original\_date\_created",  "value": "10/25/2006"  }    ],  “createParentCollections” : false,  “parentCollectionMetadataEntries” : [  {  "attribute": "collection\_type",  "value": "Folder"  }  ]  }  **XML:**  <?xml version="1.0" encoding="UTF-8" ?>  <metadataEntries>  <element>  <attribute>name</attribute>  <value>Demo Project Name10</value>  </element>  <element>  <attribute>collection\_type</attribute>  <value>Project</value>  </element>  <element>  <attribute>description</attribute>  <value>Project desc</value>  </element>  <element>  <attribute>internal\_project\_id</attribute>  <value>ipi</value>  </element>  <element>  <attribute>source\_lab\_pi</attribute>  <value>Source Lab PI</value>  </element>  <element>  <attribute>lab\_branch</attribute>  <value>Lab / Branch Name</value>  </element>  <element>  <attribute>pi\_doc</attribute>  <value>FNLCR</value>  </element>  <element>  <attribute>original\_date\_created</attribute>  <value>10/25/2006</value>  </element>  </metadataEntries> |
| Success Response | HTTP/1.1 201 Created  HPC-API-Version: 1.0.0  Location: https://hpcdmeapi.nci.nih.gov/collection/DemoProjectName  The following is the successful response on updating an existing Collection  HTTP/1.1 200 OK  HPC-API-Version: 1.0.0  Location: https://hpcdmeapi.nci.nih.gov/collection/DemoProjectName |
| Error Response | **Invalid Request Input:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  JSON:  {  "errorType": "INVALID\_REQUEST\_INPUT",  "message": "Missing mandataory metadata: name",  "stackTrace": "gov.nih.nci.hpc.exception.HpcException: Missing mandataory metadata: Project name[INVALID\_REQUEST\_INPUT]… "  }  **Invalid Request Input:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  JSON:  {  "errorType": "INVALID\_REQUEST\_INPUT",  "message": "Invalid Metadata Value: collection\_type = Project1. Valid values: [Project, Dataset, Folder]",  "stackTrace": "…"  }  **Invalid Request Input:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  JSON:  {  "errorType": "INVALID\_REQUEST\_INPUT",  "message": " Path already exists as a file:",  "stackTrace": "…"  }  **Invalid Request Input:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  JSON:  {  "errorType": "INVALID\_REQUEST\_INPUT",  "message": " System generated metadata can't be set/changed",  "stackTrace": "…"  }  **Invalid Request Input:**  This error is thrown if path is null or metadata is null or empty  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  JSON:  {  "errorType": "INVALID\_REQUEST\_INPUT",  "message": "Null path or Invalid metadata entry",  "stackTrace": "…"  }  **Invalid Request Input:**  This error is thrown if metadata entry value is missing  HTTP/1.1 500 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  JSON:  {  "errorType": "DATA\_MANAGEMENT\_ERROR",  "message": " Failed to add metadata to a collection: value is null or empty",  "stackTrace": "…"  }  **Collection Path is missing:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  JSON:  {  "errorType": "INVALID\_REQUEST\_INPUT",  "message": "Failed to create directory: /",  "stackTrace": "…"  }  **Failed to create Collection Path:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  JSON:  {  "errorType": "DATA\_MANAGEMENT\_ERROR ",  "message": "Failed to create a collection directory: /",  "stackTrace": "…"  }  **Failed to add metadata to collection:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  JSON:  {  "errorType": "DATA\_MANAGEMENT\_ERROR ",  "message": " Failed to add metadata to a collection: /",  "stackTrace": "…"  }  **Authentication Failure:**  HTTP/1.1 401 Unauthorized  Content-Type: application/json  JSON:  {  "errorType": "REQUEST\_AUTHENTICATION\_FAILED",  "message": "Access Denied: LDAP authentication failed",  "stackTrace": "…"  } |

## Find Collection by compound metadata query

|  |  |
| --- | --- |
| Title | Find collection by a compound metadata query |
| Description | First we define the simple query, then we show how to combine multiple simple queries into a compound query.  **Simple query:**  Simple query is the basic building block of the compound query. It consists of   * "attribute" * "value" * "operator" * “levelFilter” * “attributeMatch”  |  |  |  | | --- | --- | --- | | attribute | *metadata attribute name to query* | | | value | *metadata value to query* | | | operator | EQUAL, NOT\_EQUAL, LIKE, NUM\_LESS\_THAN, NUM\_LESS\_OR\_EQUAL, NUM\_GREATER\_OR\_EQUAL, NUM\_GREATER\_THAN | | | attributeMatch | ANY, EXACT | | | levelFilter | level | Metadata level filter. e.g /Coll\_A/Coll\_B/Coll\_C. The hierarchical metadata for ‘Coll\_C’ include all the metadata associated with ‘Coll\_C’ at level 1, ‘Coll\_B’ at level 2, and ‘Coll\_A’ at level 3. Please check the definition of “[Metadata Hierarchy](#_Metadata_Heirarchy)”. | | operator | Level filter operator - EQUAL, NOT\_EQUAL, NUM\_LESS\_THAN, NUM\_LESS\_OR\_EQUAL, NUM\_GREATER\_OR\_EQUAL, NUM\_GREATER\_THAN | | NOTE: ‘level’ and ‘levelOperator’ are optional. If not provided, the query will match metadata found at any level for collections search | | | | NOTE: ‘attributeMatch’ is optional. If it’s set to ANY, then the query will be apply to any metadata attribute, and in this case the ‘attribute’ of the query is expected to be omitted.  If ‘attributeMatch’ is omitted, or set to EXACT, then the search will be limited to the provided ‘attribute’ metadata. | | |   Multiple simple queries can be combined to create desired search criteria with a join operator.  **Compound Query:**  Compound query consists of three parts:   1. Join operator 2. List of queries (filtering criteria) 3. List of Compound queries (filtering criteria)   Compound query should at least have one simple query that could be part of list of queries or list of nested compound queries.  **Join Operator:**  Join operator is used to combine multiple simple or compound queries or both (i.e., simple and compound). Valid values are “AND”, “OR”.  **List of queries:**  List of queries are collection of one or more simple queries combined with join operator.  Example:   |  | | --- | | "queries":  [          {              "attribute": "name",              "value": "Experiment1",              "operator": "EQUAL",  "levelFilter": {  "level": 1,  "operator": "EQUAL"  }           },          {              "attribute": "pi\_name",              "value": "John Doe",              "operator": "EQUAL"           }       ] |   **List of compound queries:**  An optional list of compound query. The top most compound query can include a nested list of compound queries up to 10 nesting levels.  This is a nested list of compound queries to support complex search criteria.  Example:   |  | | --- | | "compoundQueries": [  {  "operator": "AND",  "queries": [  {  "attribute": "ATTR 1",  "value": "VAL 1",  "operator": "EQUAL"  },  {  “attributeMatch” : “ANY”,  "value": "VAL 2",  "operator": "LIKE"  }  ]  },  {  "operator": "OR",  "queries": [  {  "attribute": "ATTR 1",  "value": "VAL 1",  "operator": "EQUAL"  },  {  "attribute": "ATTR 2",  "value": "VAL 2",  "operator": "LIKE"  }  ]  }  ] |   detailedResponse: By default, search lists returns only path of matched resulted collections. If “detailedResponse=true”, collections are returned with metadata.  page: Ask for a specific page of results. By default, the query will return the first 100 results in page 1, the second 100 in page 2, etc. If omitted, page 1 is fetched.  totalCount: If set to ‘true’, a total count of collections matching the query regardless of the query limit and page will be returned. By default if omitted, this is set to false and no total count will be returned. |
| URL | /collection/query |
| Method | POST |
| Acceptable request representation | application/json  application/xml |
| Available response representation | application/json  application/xml |
| URL Params |  |
| Data Params | {  "compoundQuery": {  "operator": "AND",  "queries": [  {  "attribute": "ATTR 10",  "value": "VAL 1",  "operator": "EQUAL",  "levelFilter": {  "level": 1,  "operator": "NOT\_EQUAL"  }  },  {  "attribute": "ATTR 20",  "value": "VAL 2",  "operator": "LIKE"  }  ],  "compoundQueries": [  {  "operator": "AND",  "queries": [  {  "attribute": "ATTR 1",  "value": "VAL 1",  "operator": "EQUAL"  },  {  "attribute": "ATTR 2",  "value": "VAL 2",  "operator": "LIKE"  }  ]  },  {  "operator": "OR",  "queries": [  {  "attribute": "ATTR 1",  "value": "VAL 1",  "operator": "EQUAL"  },  {  "attribute": "ATTR 2",  "value": "VAL 2",  "operator": "LIKE"  }  ]  }  ]  },  "detailedResponse": false,  "page": 1,  "totalCount": false  } |
| Success Response | HTTP/1.1 200 OK  Content-Type: application/json  HPC-API-Version: 1.0.0  {"collectionPaths": [     "/Coll\_A/Coll\_B/Coll\_C”  ],  “page” : 1,  “limit” : 100,  “totalCount” : 100  } |
| Error Response | **Data Management Error:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  JSON:  {  "errorType": " DATABASE\_ERROR",  "message": " Failed to get Collections: ",  "stackTrace": "…"  }  **Invalid query attributes:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  {"errorType":"INVALID\_REQUEST\_INPUT","message":"Null compound query","stackTrace":"…"}  {"errorType":"INVALID\_REQUEST\_INPUT","message":"Null compound query operator in query","stackTrace":"…"}  {"errorType":"INVALID\_REQUEST\_INPUT","message":"Compound query contains no sub queries (simple or compound)","stackTrace":"…"}  {"errorType":"INVALID\_REQUEST\_INPUT","message":"Null metadata attribute in query","stackTrace":"…"}  {"errorType":"INVALID\_REQUEST\_INPUT","message":"Null metadata value in query","stackTrace":"…"}  {"errorType":"INVALID\_REQUEST\_INPUT","message":"Null operator in query","stackTrace":"…"}  {"errorType":"INVALID\_REQUEST\_INPUT","message":"Compound query depth over the allowed limit","stackTrace":"…"}  **No matching results:**  HTTP/1.1 204 No Content  Content-Length: 0  HPC-API-Version: 1.0.0 |

## Find Collection by compound metadata query name

|  |  |
| --- | --- |
| Title | Find collection by a compound metadata query name |
| Description | Perform search using a saved query |
| URL | /collection/query/{queryName} |
| Method | GET |
| Acceptable request representation | application/json  application/xml |
| Available response representation | application/json  application/xml |
| URL Params | queryName: The query name.  detailedResponse: By default, search lists returns only path of matched resulted collections. If “detailedResponse=true”, collections are returned with metadata.  page: Ask for a specific page of results. By default, the query will return the first 100 results in page 1, the second 100 in page 2, etc. If omitted, page 1 is fetched.  totalCount: If set to ‘true’, a total count of collections matching the query regardless of the query limit and page will be returned. By default if omitted, this is set to false and no total count will be returned. |
| Data Params |  |
| Success Response | HTTP/1.1 200 OK  Content-Type: application/json  HPC-API-Version: 1.0.0  {"collectionPaths": [     "/Coll\_A/Coll\_B/Coll\_C”  ],  “page” : 1,  “limit” : 100,  “totalCount” : 100  } |
| Error Response | **Data Management Error:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  JSON:  {  "errorType": " DATABASE\_ERROR",  "message": " Failed to get Collections: ",  "stackTrace": "…"  }  **Invalid query attributes:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  {"errorType":"INVALID\_REQUEST\_INPUT","message":"Null compound query","stackTrace":"…"}  {"errorType":"INVALID\_REQUEST\_INPUT","message":"Null compound query operator in query","stackTrace":"…"}  {"errorType":"INVALID\_REQUEST\_INPUT","message":"Compound query contains no sub queries (simple or compound)","stackTrace":"…"}  {"errorType":"INVALID\_REQUEST\_INPUT","message":"Null metadata attribute in query","stackTrace":"…"}  {"errorType":"INVALID\_REQUEST\_INPUT","message":"Null metadata value in query","stackTrace":"…"}  {"errorType":"INVALID\_REQUEST\_INPUT","message":"Null operator in query","stackTrace":"…"}  {"errorType":"INVALID\_REQUEST\_INPUT","message":"Compound query depth over the allowed limit","stackTrace":"…"}  **No matching results:**  HTTP/1.1 204 No Content  Content-Length: 0  HPC-API-Version: 1.0.0 |

## Set Collection Permissions

|  |  |
| --- | --- |
| Title | Set Permissions of a collection |
| Description | Set users and/or groups permissions on a collection. The currently supported permissions are OWN, READ, WRITE and NONE.  Setting permissions of data file at Globus level/ data or object store level is not part of this API. This API would only set permissions at iRODS entity level so that users can access metadata and contribute to collections or data files. |
| URL | /collection/{collection-path}/acl |
| Method | POST |
| Acceptable request representation | application/json  application/xml |
| Available response representation | application/json  application/xml |
| URL Params | collection-path – The collection path to set |
| Data Params | **JSON:**  {     "userPermissions": [     {        "permission": "READ",        "userId": "konkapv"     },     {        "permission": "WRITE",        "userId": "zakigf"     }],     "groupPermissions": [     {        "permission": "OWN",        "groupName": "eran-group"     }]  } |
| Success Response | HTTP/1.1 200 OK  Content-Type: application/json  HPC-API-Version: 1.0.0  {     "userPermissionResponses":    [              {           "userId": "konkapv",           "result": true        },              {           "userId": "zakigf",           "result": true        }     ],     "groupPermissionResponses": [   {        "groupName": "eran-group",        "result": true     }]  } |
| Error Response | **Missing path in the request:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  {  "errorType": "INVALID\_REQUEST\_INPUT",  "message": "Null collection path",  "stackTrace": …  }  **Invalid UserId:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  {  "errorType": "INVALID\_REQUEST\_INPUT",  "message": "Null or empty userId in a permission request for path: /FNL\_SF\_Archive/GlobusWorld2016",  "stackTrace": "gov.nih.nci.hpc.exception.HpcException: Null or empty userId in a permission request for path: /FNL\_SF\_Archive/GlobusWorld2016[INVALID\_REQUEST\_INPUT] "  }  **Invalid Group Name:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  {  "errorType": "INVALID\_REQUEST\_INPUT",  "message": "Null or empty group name in a permission request for path: /FNL\_SF\_Archive/GlobusWorld2016",  "stackTrace": "gov.nih.nci.hpc.exception.HpcException: Null or empty groupId in a permission request for path: /FNL\_SF\_Archive/GlobusWorld2016[INVALID\_REQUEST\_INPUT] "  }  **Authentication Failure:**  HTTP/1.1 401 Unauthorized  Content-Type: application/json  {"errorType":"REQUEST\_AUTHENTICATION\_FAILED","message":"Access Denied: LDAP authentication failed","stackTrace":"..”}  JSON:  {  "errorType": "REQUEST\_AUTHENTICATION\_FAILED",  "message": "Access Denied: LDAP authentication failed",  "stackTrace": "…"  } |

## Get Collection Permissions

|  |  |
| --- | --- |
| Title | GetPermissions of a collection |
| Description | Get users and groups permissions on a collection. |
| URL | /collection/{collection-path}/acl |
| Method | GET |
| Acceptable request representation | application/json  application/xml |
| Available response representation | application/json  application/xml |
| URL Params | collection-path – The collection path to get |
| Data Params |  |
| Success Response | HTTP/1.1 200 OK  Content-Type: application/json  HPC-API-Version: 1.0.0  {     "userPermissions":    [              {           "permission": "OWN",           "userId": "rods"        },              {           "permission": "READ",           "userId": "konkapv"        },              {           "permission": "OWN",           "userId": "rosenbergea"        },              {           "permission": "OWN",           "userId": "ncif-hpcdm-svc"        },              {           "permission": "WRITE",           "userId": "zakigf"        }     ],     "groupPermissions": [   {        "permission": "OWN",        "groupName": "eran-group"     }]  } |
| Error Response | **Missing path in the request:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  {  "errorType": "INVALID\_REQUEST\_INPUT",  "message": "Null collection path",  "stackTrace": …  }  **Authentication Failure:**  HTTP/1.1 401 Unauthorized  Content-Type: application/json  {"errorType":"REQUEST\_AUTHENTICATION\_FAILED","message":"Access Denied: LDAP authentication failed","stackTrace":"..”}  JSON:  {  "errorType": "REQUEST\_AUTHENTICATION\_FAILED",  "message": "Access Denied: LDAP authentication failed",  "stackTrace": "…"  } |

## GET PERMISSIONS ON COLLECTIONS

|  |  |
| --- | --- |
| Title | Get permissions on collections |
| Description | Get permissions on multiple collections accounting for all users    Note: In a response from this service, a lack of permission is not represented |
| URL | /collection/acl?collectionPath=<collection-path1>&collectionPath=<collection-path2>… |
| Method | GET |
| Acceptable request representation | Not applicable, no request body |
| Available response representation | application/json  application/xml |
| URL Params | None |
| Query String Params | collectionPath – The path of one collection; may be specified more than once  to indicate multiple collections |
| Data Params | Not applicable, no request body |
| Success Response | HTTP/1.1 200 OK  Content-Type: application/json  HPC-API-Version: 1.0.0  {  "collectionPermissions": [  {  "collectionPath": "/TEST\_Archive",  "collectionPermission": [  {  "permission": "WRITE",  "subject": "dice\_user\_group\_admin",  "subjectType": "USER"  },  {  "permission": "OWN",  "subject": "dice\_user\_sys\_admin",  "subjectType": "USER"  },  {  "permission": "OWN",  "subject": "konkapv",  "subjectType": "USER"  },  {  "permission": "OWN",  "subject": "liuwy",  "subjectType": "USER"  },  {  "permission": "OWN",  "subject": "luz6",  "subjectType": "USER"  },  {  "permission": "OWN",  "subject": "ncif-hpcdm-svc",  "subjectType": "USER"  },  {  "permission": "OWN",  "subject": "rods",  "subjectType": "USER"  },  {  "permission": "OWN",  "subject": "rosenbergea",  "subjectType": "USER"  },  {  "permission": "OWN",  "subject": "zakigf",  "subjectType": "USER"  }  ]  },  {  "collectionPath": "\TEST\_Archive/Prasad\_Test2",  "collectionPermission": [  {  "permission": "WRITE",  "subject": "dice\_user\_group\_admin",  "subjectType": "USER"  },  {  "permission": "OWN",  "subject": "dice\_user\_sys\_admin",  "subjectType": "USER"  },  {  "permission": "OWN",  "subject": "konkapv",  "subjectType": "USER"  },  {  "permission": "OWN",  "subject": "luz6",  "subjectType": "USER"  },  {  "permission": "OWN",  "subject": "ncif-hpcdm-svc",  "subjectType": "USER"  },  {  "permission": "OWN",  "subject": "rods",  "subjectType": "USER"  },  {  "permission": "OWN",  "subject": "rosenbergea",  "subjectType": "USER"  },  {  "permission": "OWN",  "subject": "zakigf",  "subjectType": "USER"  }  ]  }  ]  } |
| Error Response | **Authentication Failure:**  HTTP/1.1 401 Unauthorized  Content-Type: application/json  JSON:  {  "errorType": "REQUEST\_AUTHENTICATION\_FAILED",  "message": "Access Denied: LDAP authentication failed",  "stackTrace": "…"  }  **Other Failure (after authentication successful):**  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  {  "errorType": "<type-of-error-indicator>",  "message": "<detailed-error-message>",  "stackTrace": …  } |

## GET USER PERMISSIONS ON COLLECTIONS

|  |  |
| --- | --- |
| Title | Get user permissions on collections |
| Description | Get permissions on multiple collections for a specific user    Note: In a response from this service, a lack of permission is represented as permission setting of “NONE” |
| URL | /collection/acl/user/{userId}?collectionPath=<collection-path1>&collectionPath=<collection-path2>… |
| Method | GET |
| Acceptable request representation | Not applicable, no request body |
| Available response representation | application/json  application/xml |
| URL Params | {userId} – The user ID / user name of the specific user |
| Query String Params | collectionPath – The path of one collection; set this parameter more than once  to indicate multiple collections |
| Data Params | Not applicable, no request body |
| Success Response | HTTP/1.1 200 OK  Content-Type: application/json  HPC-API-Version: 1.0.0  {  "userId": "liuwy",  "permissionsForCollections": [  {  "permission": "OWN",  "collectionPath": "/TEST\_Archive"  },  {  "permission": "NONE",  "collectionPath": "/TEST\_Archive/Prasad\_Test2"  },  {  "permission": "READ",  "collectionPath": "/TEST\_Archive/Zhengwu\_Test1"  },  {  "permission": "WRITE",  "collectionPath": "/TEST\_Archive/dice\_project100/datset100"  }  ]  } |
| Error Response | **Authentication Failure:**  HTTP/1.1 401 Unauthorized  Content-Type: application/json  JSON:  {  "errorType": "REQUEST\_AUTHENTICATION\_FAILED",  "message": "Access Denied: LDAP authentication failed",  "stackTrace": "…"  }  **Request specifies non-existing user:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  {  "errorType": "REQUEST\_REJECTED",  "requestRejectReason": "INVALID\_NCI\_ACCOUNT",  "message": "User not found: ...",  "stackTrace": "..."  }  **Other Failure (after authentication successful):**  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  {  "errorType": "<type-of-error-indicator>",  "message": "<detailed-error-message>",  "stackTrace": …  } |

## Delete Collection

|  |  |
| --- | --- |
| Title | Delete collection |
| Description | This API deletes a collection permanentaly. Collection must be empty (i.e. not containing any data object or sub collection) before deleted |
| URL | /collection/{path:.\*} |
| Method | DELETE |
| Acceptable request representation | N/A |
| Available response representation |  |
| URL Params | Path |
| Data Params |  |
| Success Response | HTTP/1.1 200 OK  HPC-API-Version: 1.0.0 |
| Error Response | **Invalid Path:**  This error is thrown if an invalid data object path is given  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  {  "errorType": "INVALID\_REQUEST\_INPUT",  "message": "Collection not found: <Path>",  "stackTrace": "…”  }  **Collection not empty:**  This error is thrown if the collection is not empty.  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  {  {  "errorType": "INVALID\_REQUEST\_INPUT",  "message": "Collection is not empty: <path>”,  "stackTrace": "…  }  **Collection permission denied:**  This error is thrown if the invoker is not the owner of the collection.  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  {  "errorType": "REQUEST\_REJECTED",  "requestRejectReason": "DATA\_OBJECT\_PERMISSION\_DENIED",  "message": "Collection can only be deleted by its owner. Your permission: READ"  } |

## Move Collection

|  |  |
| --- | --- |
| Title | Move collection |
| Description | This API moves a collection. |
| URL | /collection/{path:.\*}/move/{destinationPath} |
| Method | POST |
| Acceptable request representation | N/A |
| Available response representation |  |
| URL Params | Path – The collection path  destinationPath – The move destination path |
| Data Params |  |
| Success Response | HTTP/1.1 200 OK  HPC-API-Version: 1.0.0 |
| Error Response | **Invalid Source Path:**  This error is thrown if an invalid source collection path is given  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  {  "errorType": "INVALID\_REQUEST\_INPUT",  "message": "Source path doesn’t exist",  "stackTrace": "…”  } |

## Register Data File

|  |  |
| --- | --- |
| Title | Register data file |
| Description | Register data file along with its metadata into DME. Registering a data file requires a registrar and PI to have a valid account with the DME. Registering a data file would transfer dataset from its origin endpoint or file system to a target archive storage.  Registering data file supports three data transfer options:   1. Synchronous data transfer from local file system to HPC Data Archive storage. 2. Asynchronous data transfer from a Globus endpoint to HPC Data Archive storage. HPC DME uses service accout to transfer data from your source location. You need to share data with “ncif-hpcdm-svc” before submitting your request. 3. User upload data using a pre-signed URL generated by the system.   Based on the given request input, the data transfer type is determined by the API.  Data file registration is done via HTTP multipart request to HPC REST interface. The following attachments are expected:   1. Attach metadata to the request (multipart)    1. ContentType: application/json    2. ContentID: dataObjectRegistration 2. Attach the file to the request (multipart) (Not needed for Globus source)    1. ContentType: application/octet-stream   ContentID: dataObject    If your data file source is a Globus endpoint location, “dataObject” attachment should not be submitted as part of the request.  If your data file source is a local file system, “source” element should be there in “dataObjectRegistration” JSON/XML submitted as part of the request.  If you want the system to generate a pre-signed URL, then “dataObjectRegistration” JSON/XML should include ‘generateUploadRequestURL’ element set to true. Both source and “dataObject” attachment should NOT be provided  By default, the service expects the containing collection for the data object to exist at time of registration. However, the user can optionally request the parent collection(s) to be created. To do so the ‘createParentCollections’ indicator needs to be set to true, and a list of metadata to create the parent collections needs to be provided. In this case the entire collection hierarchy above the data-object will be created.  For synchronous registration, caller can provide MD5 checksum value of the attached file. The server will calculate the MD5 value after receiving the file and an error will be returned if the values don’t match. |
| URL | /dataObject/{urlPath} |
| Method | PUT |
| Acceptable request representation | application/json  application/xml |
| Available response representation |  |
| URL Params | {urlPath} – Logical path of the data object. Typical structure of the dataobject can be <group folder>/<project name>/<dataset name>/<dataobject name> |
| Media Type | multipart/form-data |
| Data Params | **JSON:**  Example metadata:  {  "source": {  "fileContainerId": "e1c6b3bd-6d04-11e5-ba46-22000b92c6ec",  "fileId": "/GridFTP/GridFTP\_t3/konkapv/Data1/test2.fastq"  },  “generateUploadRequestURL” : false,  “checksum” : “d41d8cd98f00b204e9800998ecf8427e”  "metadataEntries": [  {  "attribute": "name",  "value": "Set100"  }  ],  “createParentCollections” : false,  “parentCollectionMetadataEntries” : [  {  "attribute": "collection\_type",  "value": "Folder"  }  ]  }  **XML:**  <?xml version="1.0" encoding="UTF-8" ?>  <source>  <endpoint>e1c6b3bd-6d04-11e5-ba46-22000b92c6ec</endpoint>  <path>/GridFTP/GridFTP\_t3/konkapv/Set100</path>  </source>  <filePath>/</filePath>  <metadataEntries>  <element>  <attribute>name</attribute>  <value>Set100</value>  </element>  </metadataEntries> |
| Success Response | HTTP/1.1 201 Created  Content-Length: 0  HPC-API-Version: 1.0.0  Location: <https://hpcdmeapi.nci.nih.gov/dataObject/tempZone%2Fhome%2FDemoProjectName3%2Fdata1>  Note: If a presigned upload URL was requested, a JSON/XML will be additionally returned:  {  “uploadRequestURL” : “https://…..”  } |
| Error Response | **Data file path already exists:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  JSON:  {  "errorType":"REQUEST\_REJECTED","  requestRejectReason":"DATA\_OBJECT\_PATH\_ALREADY\_EXISTS","message":"Path already exists: /tempZone/home/DemoProjectName3",  "stackTrace":"…"}  **Both data transfer source and data attachment provided:**  {  "errorType": "INVALID\_REQUEST\_INPUT",  "message": "Both data transfer source and data attachment provided",  "stackTrace": "gov.nih.nci.hpc.exception.HpcException: Both data transfer source and data attachment provided "  }  **No data transfer source or data attachment provided:**  {  "errorType": "INVALID\_REQUEST\_INPUT",  "message": "No data transfer source or data attachment provided",  "stackTrace": "gov.nih.nci.hpc.exception.HpcException: No data transfer source or data attachment provided "  }  **Data file Path is missing:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  JSON:  {  "errorType": "INVALID\_REQUEST\_INPUT",  "message": " Null path or dataObjectRegistrationDTO: /",  "stackTrace": "…"  }  **Invalid Data object Path:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  JSON:  {  "errorType": "REQUEST\_REJECTED",  "requestRejectReason": "INVALID\_DATA\_OBJECT\_PATH",  "message": "Invalid data object path. Directory doesn't exist: /FNL\_SF\_Archive/Project-test3/D/test2.fastq",  "stackTrace": "gov.nih.nci.hpc.exception.HpcException: Invalid data object path. Directory doesn't exist:..”  }  **Invalid Request Input:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  JSON:  {  "errorType": "INVALID\_REQUEST\_INPUT",  "message": "Missing mandataory metadata: Project name",  "stackTrace": "gov.nih.nci.hpc.exception.HpcException: Missing mandataory metadata: Project name[INVALID\_REQUEST\_INPUT]… "  }  **Invalid Request Input:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  JSON:  {  "errorType": "INVALID\_REQUEST\_INPUT",  "message": "Invalid Metadata Value: Collection type = Project1. Valid values: [Project, Dataset, Folder]",  "stackTrace": "…"  }  **Invalid Request Input:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  JSON:  {  "errorType": "INVALID\_REQUEST\_INPUT",  "message": "Null path or Invalid metadata entry",  "stackTrace": "…"  }  **Invalid Request Input:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  JSON:  {  "errorType": "INVALID\_REQUEST\_INPUT",  "message": "Invalid Data Transfer Input",  "stackTrace": "…"  }  **Invalid Request Input:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  JSON:  {  "errorType": "INVALID\_REQUEST\_INPUT",  "message": " Path already exists as a directory",  "stackTrace": "…"  }  **Data Management Error:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  JSON:  {  "errorType": "DATA\_MANAGEMENT\_ERROR",  "message": " Failed to create a file",  "stackTrace": "…"  }  **Data Transfer Error:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  JSON:  {  "errorType": "DATA\_TRANSFER\_ERROR",  "message": "Failed to activate endpoint: nihnci#NIH-NCI-TRANSFER",  "stackTrace": "…"  }  **Authentication Failure:**  HTTP/1.1 401 Unauthorized  Content-Type: application/json  JSON:  {  "errorType": "REQUEST\_AUTHENTICATION\_FAILED",  "message": "Access Denied: LDAP authentication failed",  "stackTrace": "…"  }  **Checksum validation failed:**  {  "errorType": "INVALID\_REQUEST\_INPUT",  "message": "Checksum validation failed",  "stackTrace": "… "  } |
|  |  |

## Bulk Data Files Registration

|  |  |
| --- | --- |
| Title | Register a list of data objects |
| Description | This API allows the user to bulk several data file registration in a single request. The user has the option to specifiy each data file registration individually, and/or request to ‘scan’ a Globus endpoint, and automatically preparing a list from the files found in that endpoint. The API allows providing a list of Globus directories to scan, as well as combining scan results w/ a list provided by the caller. The caller can provide a ‘dryRun’ indicator. If set to true, the API doesn’t perform the file registrations, but instead returns a list of the registration requests. This is useful to see the result of a Globus directory scan before committing to registering that content.  The ‘uiURL’ is optional and allows UI applications calling the API to send a URL for direct view of the registration task within the UI application. {task\_id} inside the provided URL will be replaced with actual registration task ID when the notification for completion/failures is sent. This allows for quick linkling from the notification back into the UI application.  On successful registration, the API resturns a task ID which allows the user to check progress. If the user is register to receive registration notification, a single notification will be generated when the registration is completed (either successful or failed). |
| URL | /registration |
| Method | PUT |
| Acceptable request representation | application/json  application/xml |
| Available response representation |  |
| URL Params |  |
| Media Type | Application/json |
| Data Params | **JSON:**  Example metadata:  {  "dryRun": false,  "uiURL" : "http://localhost:9080/uploadtask?taskId={task\_id}&type=",  "directoryScanRegistrationItems": [{  "basePath": "/TEST\_NO\_HIER\_Archive/eran-bulk-project",  "scanDirectoryLocation": {  "fileContainerId": "04ccf59c-815f-11e7-8dff-22000b9923ef",  "fileId": "/bulk"  },  "callerObjectId": "eran-scan-directory"  }],  "dataObjectRegistrationItems": [{  "path": "/FNL\_SF\_Archive/eran-pi-lab/eran-project/eran-flowcell/eran-sample/eran-data-object-bulk-1",  "source": {  "fileContainerId": "04ccf59c-815f-11e7-8dff-22000b9923ef",  "fileId": "/data-object"  },  "callerObjectId": "eran-base-bulk",  "metadataEntries": [{  "attribute": "name",  "value": "file-name"  },  {  "attribute": "description",  "value": "file-description"  },  {  "attribute": "source\_lab\_pi",  "value": "file-Source Lab PI"  },  {  "attribute": "lab\_branch",  "value": "file-Lab / Branch Name"  },  {  "attribute": "pi\_doc",  "value": "FNLCR"  },  {  "attribute": "flow\_cell\_id",  "value": "file-FlowCell ID"  },  {  "attribute": "run\_id",  "value": "file-Run ID"  },  {  "attribute": "run\_date",  "value": "10/11/2015"  },  {  "attribute": "library\_id",  "value": "file-Library ID"  },  {  "attribute": "test-md-1",  "value": "A"  },  {  "attribute": "test-unit",  "value": "E",  "unit": "KG"  },  {  "attribute": "object\_name",  "value": "Eran Bulk-1 object name"  }  ]  },  {  "path": "/FNL\_SF\_Archive/eran-pi-lab/eran-project/eran-flowcell/eran-sample/eran-data-object-bulk-2",  "source": {  "fileContainerId": "04ccf59c-815f-11e7-8dff-22000b9923ef",  "fileId": "/data-object"  },  "callerObjectId": "eran-base-bulk",  "metadataEntries": [{  "attribute": "name",  "value": "file-name"  },  {  "attribute": "description",  "value": "file-description"  },  {  "attribute": "source\_lab\_pi",  "value": "file-Source Lab PI"  },  {  "attribute": "lab\_branch",  "value": "file-Lab / Branch Name"  },  {  "attribute": "pi\_doc",  "value": "FNLCR"  },  {  "attribute": "flow\_cell\_id",  "value": "file-FlowCell ID"  },  {  "attribute": "run\_id",  "value": "file-Run ID"  },  {  "attribute": "run\_date",  "value": "10/11/2015"  },  {  "attribute": "library\_id",  "value": "file-Library ID"  },  {  "attribute": "test-md-1",  "value": "A"  },  {  "attribute": "test-unit",  "value": "E",  "unit": "KG"  },  {  "attribute": "object\_name",  "value": "Eran bulk-2 object name"  }  ]  }  ]  } |
| Success Response | HTTP/1.1 201 Created  Content-Length: 0  HPC-API-Version: 1.0.0 |
| Error Response | **Invalid base path:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  JSON:  {  "errorType": "INVALID\_REQUEST\_INPUT",  "message": "Can't determine configuration id for path: /TEST\_NO\_HIERD\_Archive/eran-bulk-project",  "stackTrace":"…"}  **Invalid Globus endpoint:**  {  "errorType": "DATA\_TRANSFER\_ERROR",  "message": "[GLOBUS] Endpoint doesn't exist. Make sure the endpoint name is correct and active: 104ccf59c-815f-11e7-8dff-22000b9923ef", "stackTrace": "… "  } |
|  |  |

## Find data by compound metadata query

|  |  |
| --- | --- |
| Title | Find data file by a compound metadata query |
| Description | First we define the simple query, then we show how to combine multiple simple queries into a compound query.  **Simple query:**  Simple query is the basic building block of the compound query. It consists of   * "attribute" * "value" * "operator" * “levelFilter” * “attributeMatch”  |  |  |  | | --- | --- | --- | | attribute | *metadata attribute name to query* | | | value | *metadata value to query* | | | operator | EQUAL, NOT\_EQUAL, LIKE, NUM\_LESS\_THAN, NUM\_LESS\_OR\_EQUAL, NUM\_GREATER\_OR\_EQUAL, NUM\_GREATER\_THAN | | | attributeMatch | ANY, EXACT | | | levelFilter | level | Metadata level filter. e.g /Coll\_A/Coll\_B/Coll\_C. The hierarchical metadata for ‘Coll\_C’ include all the metadata associated with ‘Coll\_C’ at level 1, ‘Coll\_B’ at level 2, and ‘Coll\_A’ at level 3. Please check the definition of “[Metadata Hierarchy](#_Metadata_Heirarchy)”. | | operator | Level filter operator - EQUAL, NOT\_EQUAL, NUM\_LESS\_THAN, NUM\_LESS\_OR\_EQUAL, NUM\_GREATER\_OR\_EQUAL, NUM\_GREATER\_THAN | | NOTE: ‘level’ and ‘levelOperator’ are optional. If not provided, the query will match metadata found at any level for collections search | | | | NOTE: ‘attributeMatch’ is optional. If it’s set to ANY, then the query will be apply to any metadata attribute, and in this case the ‘attribute’ of the query is expected to be omitted.  If ‘attributeMatch’ is omitted, or set to EXACT, then the search will be limited to the provided ‘attribute’ metadata. | | |   Multiple simple queries can be combined to create desired search criteria with a join operator.  **Compound Query:**  Compound query consists of three parts:   1. Join operator 2. List of queries (filtering criteria) 3. List of Compound queries (filtering criteria)   Compound query should at least have one simple query that could be part of list of queries or list of nested compound queries.  **Join Operator:**  Join operator is used to combine multiple simple or compound queries or both (i.e., simple and compound). Valid values are “AND”, “OR”.  **List of queries:**  List of queries are collection of one or more simple queries combined with join operator.  Example:   |  | | --- | | "queries":  [          {              "attribute": "name",              "value": "Experiment1",              "operator": "EQUAL",  "levelFilter": {  "level": 1,  "operator": "EQUAL"  }           },          {              "attribute": "pi\_name",              "value": "John Doe",              "operator": "EQUAL"           }       ] |   **List of compound queries:**  An optional list of compound query. To top most compound query can include a nested list of compound queries up to 10 nesting levels.  This is a nested list of compound queries to support complex search criteria.  Example:   |  | | --- | | "compoundQueries": [  {  "operator": "AND",  "queries": [  {  "attribute": "ATTR 1",  "value": "VAL 1",  "operator": "EQUAL"  },  {  "matchAttribute": "ANY",  "value": "%VAL%",  "operator": "LIKE"  }  ]  },  {  "operator": "OR",  "queries": [  {  "attribute": "ATTR 1",  "value": "VAL 1",  "operator": "EQUAL"  },  {  "attribute": "ATTR 2",  "value": "VAL 2",  "operator": "LIKE"  }  ]  }  ] |   detailedResponse: By default, search lists returns only path of matched resulted collections. If “detailedResponse=true”, collections are returned with metadata.  page: Ask for a specific page of results. By default, the query will return the first 100 results in page 1, the second 100 in page 2, etc. If omitted, page 1 is fetched.  totalCount: If set to ‘true’, a total count of collections matching the query regardless of the query limit and page will be returned. By default if omitted, this is set to false and no total count will be returned. |
| URL | /dataObject/query |
| Method | POST |
| Acceptable request representation | application/json  application/xml |
| Available response representation | application/json  application/xml |
| URL Params |  |
| Data Params | {  "compoundQuery":{  "operator": "AND",  "queries":[  {  "attribute": "ATTR 1",  "value": "VAL 1",  "operator": "EQUAL",  "levelFilter": {  "level": 1,  "operator": "NOT\_EQUAL"  }  },  {  "attribute": "ATTR 2",  "value": "VAL 2",  "operator": "LIKE"  }  ],  "compoundQueries":[  {              "operator": "AND",              "queries":[  {  "attribute": "ATTR 1",                   "value": "VAL 1",                   "operator": "EQUAL"              },               {               "attribute": "ATTR 2",                   "value": "VAL 2",                   "operator": "LIKE"               }              ]           },           {              "operator": "OR",              "queries":[               {                    "attribute": "ATTR 1",                    "value": "VAL 1",                    "operator": "EQUAL"                 },                 {                    "attribute": "ATTR 2",                    "value": "VAL 2",                    "operator": "LIKE"                 }              ]           }        ]     },     "detailedResponse": false,  "page": 1,  "totalCount": false,  } |
| Success Response | HTTP/1.1 200 OK  Content-Type: application/json  HPC-API-Version: 1.0.0  {"collectionPaths": [     "/Coll\_A/Coll\_B/Data\_A”  ],  “page” : 1,  “limit” : 100,  “totalCount” : 100  } |
| Error Response | **Data Management Error:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  JSON:  {  "errorType": " DATABASE\_ERROR",  "message": " Failed to get Collections: ",  "stackTrace": "…"  }  **Invalid query attributes:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  {"errorType":"INVALID\_REQUEST\_INPUT","message":"Null compound query","stackTrace":"…"}  {"errorType":"INVALID\_REQUEST\_INPUT","message":"Null compound query operator in query","stackTrace":"…"}  {"errorType":"INVALID\_REQUEST\_INPUT","message":"Compound query contains no sub queries (simple or compound)","stackTrace":"…"}  {"errorType":"INVALID\_REQUEST\_INPUT","message":"Null metadata attribute in query","stackTrace":"…"}  {"errorType":"INVALID\_REQUEST\_INPUT","message":"Null metadata value in query","stackTrace":"…"}  {"errorType":"INVALID\_REQUEST\_INPUT","message":"Null operator in query","stackTrace":"…"}  {"errorType":"INVALID\_REQUEST\_INPUT","message":"Compound query depth over the allowed limit","stackTrace":"…"}  **No matching results:**  HTTP/1.1 204 No Content  Content-Length: 0  HPC-API-Version: 1.0.0 |

## Find data by compound metadata query name

|  |  |
| --- | --- |
| Title | Find data by a compound metadata query name |
| Description | Invoke a saved query . |
| URL | /dataObject/query/{queryName} |
| Method | GET |
| Acceptable request representation | application/json  application/xml |
| Available response representation | application/json  application/xml |
| URL Params | queryName: The query name.  detailedResponse: By default, search lists returns only path of matched resulted collections. If “detailedResponse=true”, collections are returned with metadata.  page: Ask for a specific page of results. By default, the query will return the first 100 results in page 1, the second 100 in page 2, etc. If omitted, page 1 is fetched.  totalCount: If set to ‘true’, a total count of collections matching the query regardless of the query limit and page will be returned. By default if omitted, this is set to false and no total count will be returned. |
| Data Params |  |
| Success Response | HTTP/1.1 200 OK  Content-Type: application/json  HPC-API-Version: 1.0.0  {"collectionPaths": [     "/Coll\_A/Coll\_B/Data\_A”,  ],  “page” : 1,  “limit” : 100,  “totalCount” : 100  } |
| Error Response | **Data Management Error:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  JSON:  {  "errorType": " DATABASE\_ERROR",  "message": " Failed to get Collections: ",  "stackTrace": "…"  }  **Invalid query attributes:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  {"errorType":"INVALID\_REQUEST\_INPUT","message":"Null compound query","stackTrace":"…"}  {"errorType":"INVALID\_REQUEST\_INPUT","message":"Null compound query operator in query","stackTrace":"…"}  {"errorType":"INVALID\_REQUEST\_INPUT","message":"Compound query contains no sub queries (simple or compound)","stackTrace":"…"}  {"errorType":"INVALID\_REQUEST\_INPUT","message":"Null metadata attribute in query","stackTrace":"…"}  {"errorType":"INVALID\_REQUEST\_INPUT","message":"Null metadata value in query","stackTrace":"…"}  {"errorType":"INVALID\_REQUEST\_INPUT","message":"Null operator in query","stackTrace":"…"}  {"errorType":"INVALID\_REQUEST\_INPUT","message":"Compound query depth over the allowed limit","stackTrace":"…"}  **No matching results:**  HTTP/1.1 204 No Content  Content-Length: 0  HPC-API-Version: 1.0.0 |

## Download data file

|  |  |
| --- | --- |
| Title | Download/transfer data file from Archive storage to another location |
| Description | This API transfers a data file from archive storage to another accessible location, which may be a Globus endpoint to the requestor or their own accessible file system. Data object path represents the logical path of the data object registered with HPC DME. Destination path in the request represents the physical path on the Globus endpoint. When a user requests a download, the API lets user keep file/folder name same as it is at the source or change it. If you are downloading a file and if the destination path is a folder, the file will be downloaded into the given folder. If the destination path is a file, given destination file name will be used to save the file.  An optional ‘generateDownloadRequestURL’ indicator can be provided, and it is set to ‘true’, a presigned URL will be returned. Users can use GET operatior on the presigned URL to download file directly from the archive. This step would bypass the need to stream the data file from the archive to API server and then to the client. This option would only work for synchronous data file download. ‘destination’ should not be specified with this option.  An optional ‘destinationOverwrite’ indicator can be provided, and if set to ‘true’ and the destination file exists, it will get overwritten. If this indicator is omitted, then the transfer request will get rejected if the destination file exists.  Note: Write access right needs to be granted to the service account if a user doanloads a data file from Archive to a Globus end point |
| URL | /dataObject/{path:.\*}/download |
| Method | POST |
| Acceptable request representation | application/json  application/xml |
| Available response representation |  |
| URL Params | path |
| Data Params | **To request a download to a Globus endpoint**  **JSON:**  {  "destination": {  "fileContainerId": "nihnci#NIH-NCI-TRANSFER1",  "fileId": "/GridFTP/GridFTP\_t3/konkapv/Set100"  },  "destinationOverwrite" : false  }  **XML:**  <?xml version="1.0" encoding="UTF-8" ?>  < destination>  <fileContainerId>nihnci#NIH-NCI-TRANSFER1</endpoint>  <fileId>/GridFTP/GridFTP\_t3/konkapv/Set100</path>  </ destination>  **To request a direct data download from Cleversafe**  **JSON:**  **{**  **}**  **XML**  <?xml version="1.0" encoding="UTF-8" ?>  **To request a presigned URL**  **JSON:**  {  generateDownloadRequestURL:"true"  }  **XML**  <?xml version="1.0" encoding="UTF-8" ?>  < generateDownloadRequestURL>true</ generateDownloadRequestURL> |
| Success Response | **Globus download response**  HTTP/1.1 200 OK  Content-Type: application/json  HPC-API-Version: 1.0.0  {     "taskId": 4,     "destination":    {        "fileContainerId": "eranrosenberg#hpc-test",        "fileId": "/~/Development/Tools/globus/drop/eran-data-object-file"     }  }  **Cleversafe data download response**  <data contentType="application/octet-stream" contentLength="562728">  -- Binary Data – |
| Error Response | **Invalid Path:**  This error is thrown if an invalid data object path is given  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  {  "errorType": "INVALID\_REQUEST\_INPUT",  "message": "Data object not found: <Path>",  "stackTrace": "gov.nih.nci.hpc.exception.HpcException: Data object not found: /FNL\_SF\_Archive/projectx1/data[INVALID\_REQUEST\_INPUT]”  }  **File is not archived yet:**  This error is thrown if the file is not in ARCHIVED state.  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  {  "errorType": "REQUEST\_REJECTED",  "requestRejectReason": "FILE\_NOT\_ARCHIVED",  "message": "Object is not in archived state yet. It is in <IN\_PROGRESS\_TO\_TEMPORARY\_ARCHIVE> state",  "stackTrace": "gov.nih.nci.hpc.exception.HpcException: Object is not in archived state yet. It is in IN\_PROGRESS\_TO\_TEMPORARY\_ARCHIVE state[REQUEST\_REJECTED]”  }  **Invalid Data transfer request:**  This error is thrown if an invalid Globus endpoint address is given  HTTP/1.1 500 Server ErrorContent-Type: application/json  HPC-API-Version: 1.0.0  {  "errorType": "DATA\_TRANSFER\_ERROR",  "message": "Failed to activate endpoint: nihfnlcr#gridftp",  "stackTrace": "gov.nih.nci.hpc.exception.HpcException: Failed to activate endpoint: nihfnlcr#gridftp[DATA\_TRANSFER\_ERROR]”  }  **Invalid Data transfer request:**  This error is thrown if Globus endpoint address(fileContainerId) or fileId is missing in the request  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  {  "errorType": "INVALID\_REQUEST\_INPUT",  "message": "Invalid file location",  "stackTrace": "gov.nih.nci.hpc.exception.HpcException: Invalid file location[INVALID\_REQUEST\_INPUT]”  }  **Invalid Data transfer request:**  This error is thrown if generateDownloadRequestURL option is set to true on POSIX file system based archive  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  {  "errorType": "INVALID\_REQUEST\_INPUT",  "message": "Presigned URL for download is supported on S3 based destination archive only. Requested path is archived on a POSIX based file system",  "stackTrace": "gov.nih.nci.hpc.exception.HpcException: Presigned URL for download is supported on S3 based destination archive only. Requested path is archived on a POSIX based file system [INVALID\_REQUEST\_INPUT]”  }  **Invalid Data transfer request:**  This error is thrown if both destination and generateDownloadRequestURL’ is mentioned in the request  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  {  "errorType": "INVALID\_REQUEST\_INPUT",  "message": " Both data transfer destination and Presigned URL request provided ",  "stackTrace": "gov.nih.nci.hpc.exception.HpcException: Both data transfer destination and Presigned URL request provided [INVALID\_REQUEST\_INPUT]”  } |

## Get data file Download Status

|  |  |
| --- | --- |
| Title | Get a status of data file download task |
| Description | When a download request is submitted with a Globus destination, the service returns a ‘taskId’ to the caller immediately and the download is performed asynchronously. This service allows the user to check the status of the download task |
| URL | /dataObject/download?taskId=<id> |
| Method | GET |
| Acceptable request representation | application/json  application/xml |
| Available response representation |  |
| URL Params | taskId The task ID received when submitted the download request |
| Data Params |  |
| Success Response | HTTP/1.1 200 OK  Content-Type: application/json  HPC-API-Version: 1.0.0  {  "inProgress": false,  "path": "/FNL\_SF\_Archive/eran-pi-lab/eran-project/eran-flowcell/eran-sample/eran-data-object-s3",  "dataTransferRequestId": "222f102e-6d56-11e7-a9cd-22000bf2d287",  "dataTransferType": "GLOBUS",  "destinationLocation": {  "fileContainerId": "eranrosenberg#eran-share",  "fileId": "/download/col-test-new/eran-project/eran-flowcell/eran-sample/eran-data-object-s3"  },  "result": true,  "created": "2017-07-20T10:17:32.349-04:00",  "completed": "2017-07-20T16:43:03.830-04:00"  } |
| Error Response | **Invalid TaskId:**  This error is thrown if an invalid data object path is given  HTTP/1.1 204 No Content  Content-Type: application/json  HPC-API-Version: 1.0.0 |

## Download Collection

|  |  |
| --- | --- |
| Title | Download/transfer data file from Archive storage to another location |
| Description | This API transfers a data file from archive storage to another accessible Globus location. Path represents the logical path of the collection registered with HPC DME. Destination path in the request represents the physical path on the Globus endpoint.  An optional ‘destinationOverwrite’ indicator can be provided, and if set to ‘true’ and the destination file exists, it will get overwritten. If this indicator is omitted, then the transfer request will get rejected if the destination file exists.  Note: Write access right needs to be granted to the service account if a user doanloads a data file from Archive to a Globus end point |
| URL | /collection/{path:.\*}/download |
| Method | POST |
| Acceptable request representation | application/json  application/xml |
| Available response representation |  |
| URL Params | path |
| Data Params | **To request a download to a Globus endpoint**  **JSON:**  {  "destination": {  "fileContainerId": "nihnci#NIH-NCI-TRANSFER1",  "fileId": "/GridFTP/GridFTP\_t3/konkapv/Set100"  },  destinationOverwrite" : false,  }  **XML:**  <?xml version="1.0" encoding="UTF-8" ?>  < destination>  <fileContainerId>nihnci#NIH-NCI-TRANSFER1</endpoint>  <fileId>/GridFTP/GridFTP\_t3/konkapv/Set100</path>  </ destination> |
| Success Response | HTTP/1.1 200 OK  Content-Type: application/json  HPC-API-Version: 1.0.0  {     "taskId": 4,     "destination":    {        "fileContainerId": "eranrosenberg#hpc-test",        "fileId": "/~/Development/Tools/globus/drop/eran-data-object-file"     }  } |
| Error Response | **Invalid Path:**  This error is thrown if an invalid data object path is given  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  {  "errorType": "INVALID\_REQUEST\_INPUT",  "message": "Collection not found: <Path>",  "stackTrace": "gov.nih.nci.hpc.exception.HpcException: Data object not found: /FNL\_SF\_Archive/projectx1/data[INVALID\_REQUEST\_INPUT]”  } |

## Get Collection Download Status

|  |  |
| --- | --- |
| Title | Get a status of data file download task |
| Description | When a download request is submitted with a Globus destination, the service returns a ‘taskId’ to the caller immediately and the download is performed asynchronously. This service allows the user to check the status of the download task |
| URL | /collection/download?taskId=<id> |
| Method | GET |
| Acceptable request representation | application/json  application/xml |
| Available response representation |  |
| URL Params | taskId The task ID received when submitted the download request |
| Data Params |  |
| Success Response | HTTP/1.1 200 OK  Content-Type: application/json  HPC-API-Version: 1.0.0  {  "inProgress": false,  "path": "/FNL\_SF\_Archive/eran-pi-lab",  "destinationLocation": {  "fileContainerId": "eranrosenberg#eran-share",  "fileId": "/download/api-doc-col"  },  "result": true,  "completedItems": [  {  "dataObjectDownloadTaskId": 5,  "path": "/FNL\_SF\_Archive/eran-pi-lab/eran-project/eran-flowcell/eran-sample/eran-data-object-globus",  "destinationLocation": {  "fileContainerId": "eranrosenberg#eran-share",  "fileId": "/download/api-doc-col/eran-project/eran-flowcell/eran-sample/eran-data-object-globus"  },  "result": true  },  {  "dataObjectDownloadTaskId": 6,  "path": "/FNL\_SF\_Archive/eran-pi-lab/eran-project/eran-flowcell/eran-sample/eran-data-object-s3",  "destinationLocation": {  "fileContainerId": "eranrosenberg#eran-share",  "fileId": "/download/api-doc-col/eran-project/eran-flowcell/eran-sample/eran-data-object-s3"  },  "result": true  },  {  "dataObjectDownloadTaskId": 7,  "path": "/FNL\_SF\_Archive/eran-pi-lab/eran-project/eran-flowcell/eran-sample/eran-data-object-s3-test-old-dm",  "destinationLocation": {  "fileContainerId": "eranrosenberg#eran-share",  "fileId": "/download/api-doc-col/eran-project/eran-flowcell/eran-sample/eran-data-object-s3-test-old-dm"  },  "result": true  }  ],  "created": "2017-07-20T16:37:55.005-04:00",  "completed": "2017-07-20T16:53:00.234-04:00"  } |
| Error Response | **Invalid TaskId:**  This error is thrown if an invalid data object path is given  HTTP/1.1 204 No Content  Content-Type: application/json  HPC-API-Version: 1.0.0 |

## Download Data Object List

|  |  |
| --- | --- |
| Title | Download/transfer data file from Archive storage to another location |
| Description | This API transfers a data file from archive storage to another accessible Globus location. The caller provides a list of data object paths. Destination path in the request represents the physical path on the Globus endpoint.  An optional ‘destinationOverwrite’ indicator can be provided, and if set to ‘true’ and the destination file exists, it will get overwritten. If this indicator is omitted, then the transfer request will get rejected if the destination file exists.  Note: Write access right needs to be granted to the service account if a user doanloads a data file from Archive to a Globus end point |
| URL | /download |
| Method | POST |
| Acceptable request representation | application/json  application/xml |
| Available response representation |  |
| URL Params | path |
| Data Params | **To request a download to a Globus endpoint**  {  "dataObjectPaths" : [  "/FNL\_SF\_Archive/eran-pi-lab/eran-project/eran-flowcell/eran-sample/eran-data-object-globus",  "/FNL\_SF\_Archive/eran-pi-lab/eran-project/eran-flowcell/eran-sample/eran-data-object-s3"  ],  "destination": {  "fileContainerId" : "4a3b132a-815f-11e7-8dff-22000b9923ef",  "fileId" : "/my-folder"  },  destinationOverwrite" : false,  } |
| Success Response | HTTP/1.1 200 OK  Content-Type: application/json  HPC-API-Version: 1.0.0  {  "taskId": "7a76bb60-628e-4f58-b24b-41984eea276f",  "destinationLocation": {  "fileContainerId": "4a3b132a-815f-11e7-8dff-22000b9923ef",  "fileId": "/my-folder"  }  } |
| Error Response | **Invalid Path:**  This error is thrown if an invalid data object path is given  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  {  "errorType": "INVALID\_REQUEST\_INPUT",  "message": "Data Object not found: <Path>",  } |

## Get Data Object List Download Status

|  |  |
| --- | --- |
| Title | Get a status of data file download task |
| Description | When a download request is submitted with a Globus destination, the service returns a ‘taskId’ to the caller immediately and the download is performed asynchronously. This service allows the user to check the status of the download task |
| URL | /download?taskId=<id> |
| Method | GET |
| Acceptable request representation | application/json  application/xml |
| Available response representation |  |
| URL Params | taskId The task ID received when submitted the download request |
| Data Params |  |
| Success Response | HTTP/1.1 200 OK  Content-Type: application/json  HPC-API-Version: 1.0.0  {  "inProgress": false,  "destinationLocation": {  "fileContainerId": "4a3b132a-815f-11e7-8dff-22000b9923ef",  "fileId": "/test-downloading-objs-listtt"  },  "result": true,  "completedItems": [  {  "dataObjectDownloadTaskId": "5dd1de0d-927b-4807-a3cb-8c992d8d8d01",  "path": "/FNL\_SF\_Archive/eran-pi-lab/eran-project/eran-flowcell/eran-sample/eran-data-object-globus",  "destinationLocation": {  "fileContainerId": "4a3b132a-815f-11e7-8dff-22000b9923ef",  "fileId": "/test-downloading-objs-listtt/eran-data-object-globus"  },  "result": true  },  {  "dataObjectDownloadTaskId": "16569442-5d45-492a-9378-1e9a4f33fcf1",  "path": "/FNL\_SF\_Archive/eran-pi-lab/eran-project/eran-flowcell/eran-sample/eran-data-object-s3",  "destinationLocation": {  "fileContainerId": "4a3b132a-815f-11e7-8dff-22000b9923ef",  "fileId": "/test-downloading-objs-listtt/eran-data-object-s3"  },  "result": true  }  ],  "created": "2017-08-15T22:15:55.005-04:00",  "completed": "2017-08-15T22:19:06.780-04:00"  } |
| Error Response | **Invalid TaskId:**  This error is thrown if an invalid data object path is given  HTTP/1.1 204 No Content  Content-Type: application/json  HPC-API-Version: 1.0.0 |

## Delete data file

|  |  |
| --- | --- |
| Title | Delete data file from Archive storage. |
| Description | This API deletes a data object permanentaly. |
| URL | /dataObject/{path:.\*} |
| Method | DELETE |
| Acceptable request representation | N/A |
| Available response representation |  |
| URL Params | path |
| Data Params |  |
| Success Response | HTTP/1.1 200 OK  HPC-API-Version: 1.0.0  {  "dataManagementDeleteStatus": true,  "archiveDeleteStatus": true  } |
| Error Response | **Invalid Path:**  This error is thrown if an invalid data object path is given  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  {  "errorType": "INVALID\_REQUEST\_INPUT",  "message": "Data object not found: <Path>",  "stackTrace": "gov.nih.nci.hpc.exception.HpcException: Data object not found: /FNL\_SF\_Archive/projectx1/data[INVALID\_REQUEST\_INPUT]”  }  **File is not archived yet:**  This error is thrown if the file is not in ARCHIVED state.  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  {  "errorType": "REQUEST\_REJECTED",  "requestRejectReason": "FILE\_NOT\_ARCHIVED",  "message": "Object is not in archived state yet. It is in <IN\_PROGRESS\_TO\_TEMPORARY\_ARCHIVE> state",  "stackTrace": "gov.nih.nci.hpc.exception.HpcException: Object is not in archived state yet. It is in IN\_PROGRESS\_TO\_TEMPORARY\_ARCHIVE state[REQUEST\_REJECTED]”  }  **Data object permission denied:**  This error is thrown if the invoker is not the owner of the file.  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  {  "errorType": "REQUEST\_REJECTED",  "requestRejectReason": "DATA\_OBJECT\_PERMISSION\_DENIED",  "message": "Data object can only be deleted by its owner. Your permission: READ"  } |

## Move data file

|  |  |
| --- | --- |
| Title | Move data file. |
| Description | This API move a data file |
| URL | /dataObject/{path:.\*}/move/{destinationPath} |
| Method | POST |
| Acceptable request representation | N/A |
| Available response representation |  |
| URL Params | path – the data file path  destinationPath– the move destination path |
| Data Params |  |
| Success Response | HTTP/1.1 200 OK  HPC-API-Version: 1.0.0 |
| Error Response | **Invalid Source Path:**  This error is thrown if an invalid source data object path is given  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  {  "errorType": "INVALID\_REQUEST\_INPUT",  "message": "Source path doesn’t exist",  } |

## Set Data Permissions

|  |  |
| --- | --- |
| Title | Set Permissions of a collection |
| Description | Set users and/or groups permissions on a data object. The currently supported permissions are OWN, READ, WRITE and NONE.  Setting permissions of data file at Globus level/ data or object store level is not part of this API. This API would only set permissions at iRODS entity level so that users can access metadata and contribute to collections or data files. |
| URL | /dataObject/{data-object-path}/acl |
| Method | POST |
| Acceptable request representation | application/json  application/xml |
| Available response representation | application/json  application/xml |
| URL Params | Data-object-path – The data-object path to set |
| Data Params | **JSON:**  {     "userPermissions": [     {        "permission": "READ",        "userId": "konkapv"     },     {        "permission": "WRITE",        "userId": "zakigf"     }],     "groupPermissions": [     {        "permission": "OWN",        "groupName": "eran-group"     }]  } |
| Success Response | HTTP/1.1 200 OK  Content-Type: application/json  HPC-API-Version: 1.0.0  {     "userPermissionResponses":    [              {           "userId": "konkapv",           "result": true        },              {           "userId": "zakigf",           "result": true        }     ],     "groupPermissionResponses": [   {        "groupName": "eran-group",        "result": true     }]  } |
| Error Response | **Missing path in the request:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  {  "errorType": "INVALID\_REQUEST\_INPUT",  "message": "Null data-object path",  "stackTrace": …  }  **Invalid UserId:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  {  "errorType": "INVALID\_REQUEST\_INPUT",  "message": "Null or empty userId in a permission request for path: /FNL\_SF\_Archive/GlobusWorld2016",  "stackTrace": "gov.nih.nci.hpc.exception.HpcException: Null or empty userId in a permission request for path: /FNL\_SF\_Archive/GlobusWorld2016[INVALID\_REQUEST\_INPUT] "  }  **Invalid Group Name:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  {  "errorType": "INVALID\_REQUEST\_INPUT",  "message": "Null or empty group name in a permission request for path: /FNL\_SF\_Archive/GlobusWorld2016",  "stackTrace": "gov.nih.nci.hpc.exception.HpcException: Null or empty groupId in a permission request for path: /FNL\_SF\_Archive/GlobusWorld2016[INVALID\_REQUEST\_INPUT] "  }  **Authentication Failure:**  HTTP/1.1 401 Unauthorized  Content-Type: application/json  {"errorType":"REQUEST\_AUTHENTICATION\_FAILED","message":"Access Denied: LDAP authentication failed","stackTrace":"..”}  JSON:  {  "errorType": "REQUEST\_AUTHENTICATION\_FAILED",  "message": "Access Denied: LDAP authentication failed",  "stackTrace": "…"  } |

## Get Data Permissions

|  |  |
| --- | --- |
| Title | GetPermissions of a data object |
| Description | Get users and groups permissions on a data object. |
| URL | /dataObject/{data-object-path}/acl |
| Method | GET |
| Acceptable request representation | application/json  application/xml |
| Available response representation | application/json  application/xml |
| URL Params | Data-object-path – The data-object path to get |
| Data Params |  |
| Success Response | HTTP/1.1 200 OK  Content-Type: application/json  HPC-API-Version: 1.0.0  {     "userPermissions":    [              {           "permission": "OWN",           "userId": "rods"        },              {           "permission": "READ",           "userId": "konkapv"        },              {           "permission": "OWN",           "userId": "rosenbergea"        },              {           "permission": "OWN",           "userId": "ncif-hpcdm-svc"        },              {           "permission": "WRITE",           "userId": "zakigf"        }     ],     "groupPermissions": [   {        "permission": "OWN",        "groupName": "eran-group"     }]  } |
| Error Response | **Missing path in the request:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  {  "errorType": "INVALID\_REQUEST\_INPUT",  "message": "Null data-object path",  "stackTrace": …  }  **Authentication Failure:**  HTTP/1.1 401 Unauthorized  Content-Type: application/json  {"errorType":"REQUEST\_AUTHENTICATION\_FAILED","message":"Access Denied: LDAP authentication failed","stackTrace":"..”}  JSON:  {  "errorType": "REQUEST\_AUTHENTICATION\_FAILED",  "message": "Access Denied: LDAP authentication failed",  "stackTrace": "…"  } |

## Subscribe to notifications

|  |  |
| --- | --- |
| Title | Subscribe to event notifications |
| Description | Authorized users can subscribe or unsubscribe to the following HPC DME event notifications:  DATA\_TRANSFER\_UPLOAD\_IN\_TEMPORARY\_ARCHIVE:  DATA\_TRANSFER\_UPLOAD\_ARCHIVED:  DATA\_TRANSFER\_UPLOAD\_FAILED:  DATA\_TRANSFER\_DOWNLOAD\_COMPLETED:  DATA\_TRANSFER\_DOWNLOAD\_FAILED:  USAGE\_SUMMARY\_REPORT:  USAGE\_SUMMARY\_BY\_WEEKLY\_REPORT:  COLLECTION\_UPDATED  Add/Update subscriptions should be included in the ‘addUpdateSubscriptions’ array, and delete subscriptions events should be included in the ‘deleteSubscriptions’. When you are subscribing to COLLECTION\_UPDATED notification, by default, any update within the entire collection tree (including all children) will generate a notification. |
| URL | /notification |
| Method | POST |
| Acceptable request representation | application/json  application/xml |
| Available response representation | application/json  application/xml |
| URL Params |  |
| Data Params | **JSON:**  {  "addUpdateSubscriptions": [  {  "eventType": "DATA\_TRANSFER\_UPLOAD\_ARCHIVED",  "notificationDeliveryMethods": ["EMAIL"]  },  {  "eventType": "DATA\_TRANSFER\_UPLOAD\_IN\_TEMPORARY\_ARCHIVE",  "notificationDeliveryMethods": ["EMAIL"]  },  {  "eventType": "DATA\_TRANSFER\_UPLOAD\_FAILED",  "notificationDeliveryMethods": ["EMAIL"]  },  {  "eventType": "DATA\_TRANSFER\_DOWNLOAD\_COMPLETED",  "notificationDeliveryMethods": ["EMAIL"]  },  {  "eventType": "DATA\_TRANSFER\_DOWNLOAD\_FAILED",  "notificationDeliveryMethods": ["EMAIL"]  },  {  "eventType": "USAGE\_SUMMARY\_REPORT",  "notificationDeliveryMethods": ["EMAIL"]  },  {  "eventType": "USAGE\_SUMMARY\_BY\_WEEKLY\_REPORT",  "notificationDeliveryMethods": ["EMAIL"]  },  {  "eventType" : "COLLECTION\_UPDATED",  "notificationDeliveryMethods" : ["EMAIL"],  "notificationTriggers" : [  { "payloadEntries" : [  { "attribute" : "COLLECTION\_PATH",  "value" : "/FNL\_SF\_Archive/pi-lab/project"  }  ]  },  { "payloadEntries" : [  { "attribute" : "COLLECTION\_PATH",  "value" : "/FNL\_SF\_Archive/pi-lab"  }  ]  }  ]  }  ],  "deleteSubscriptions": [“DATA\_TRANSFER\_UPLOAD\_ARCHIVED”, “USAGE\_SUMMARY\_REPORT”]  } |
| Success Response | HTTP/1.1 200 OK  Content-Type: application/json  HPC-API-Version: 1.0.0 |
| Error Response | **Invaid notification type:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  {  "errorType": "INVALID\_REQUEST\_INPUT",  "message": "Invalid add/update notification subscription request",  "stackTrace": ".."  }  **Authentication Failure:**  HTTP/1.1 401 Unauthorized  Content-Type: application/json  {"errorType":"REQUEST\_AUTHENTICATION\_FAILED","message":"Access Denied: LDAP authentication failed","stackTrace":"..”}  JSON:  {  "errorType": "REQUEST\_AUTHENTICATION\_FAILED",  "message": "Access Denied: LDAP authentication failed",  "stackTrace": "…"  } |

## Get notification subscription

|  |  |
| --- | --- |
| Title | Get notification subscriptions of a user |
| Description | Users can check what notifications they are subscribed to |
| URL | /notification |
| Method | GET |
| Acceptable request representation | application/json  application/xml |
| Available response representation | application/json  application/xml |
| URL Params |  |
| Data Params |  |
| Success Response | HTTP/1.1 200 OK  Access-Control-Allow-Origin: \*  Access-Control-Allow-Methods: GET, POST, DELETE, PUT  HPC-API-Version: 1.0.0  Content-Type: application/json  Date: Thu, 25 Aug 2016 01:02:02 GMT  Transfer-Encoding: chunked  Server: Jetty(8.1.17.v20150415)  {"subscriptions": [  {  "eventType": "DATA\_TRANSFER\_DOWNLOAD\_COMPLETED",  "notificationDeliveryMethods": "EMAIL"  },  {  "eventType": "DATA\_TRANSFER\_UPLOAD\_IN\_TEMPORARY\_ARCHIVE",  "notificationDeliveryMethods": "EMAIL"  },  {  "eventType": "DATA\_TRANSFER\_UPLOAD\_ARCHIVED",  "notificationDeliveryMethods": "EMAIL"  },  {  "eventType": "DATA\_TRANSFER\_UPLOAD\_FAILED",  "notificationDeliveryMethods": "EMAIL"  },  {  "eventType": "DATA\_TRANSFER\_DOWNLOAD\_FAILED",  "notificationDeliveryMethods": "EMAIL"  }  ]} |
| Error Response | **Invaid notification user ID:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  Date: Thu, 25 Aug 2016 01:03:29 GMT  Transfer-Encoding: chunked  Server: Jetty(8.1.17.v20150415)  {  "errorType": "INVALID\_REQUEST\_INPUT",  "message": "Invalid user: <user>  } |

## Get User notifications

|  |  |
| --- | --- |
| Title | Get User notifications |
| Description | Get User notifications generated by HPC DME with pagination support. Response should include current page number, total count, page size. |
| URL | /hpc-server/notification/deliveryReceipts |
| Method | GET |
| Acceptable request representation | application/json  application/xml |
| Available response representation | application/json  application/xml |
| URL Params | page – The results page to get. If omiited, page 1 is the default  totalCount – Request the service to return total count of notifications. Valid values are ‘true’ and false. If omitted, ‘false’ is the default |
| Data Params |  |
| Success Response | HTTP/1.1 200 OK  Access-Control-Allow-Origin: \*  Access-Control-Allow-Methods: GET, POST, DELETE, PUT  HPC-API-Version: 1.0.0  Content-Type: application/json  Date: Thu, 25 Aug 2016 01:02:02 GMT  Transfer-Encoding: chunked  Server: Jetty(8.1.17.v20150415)  {     "notificationDeliveryReceipts":    [              {           "eventId": 4,           "eventType": "DATA\_TRANSFER\_DOWNLOAD\_COMPLETED",           "eventPayloadEntries": [         {              "attribute": "DATA\_TRANSFER\_REQUEST\_ID",              "value": "cd7aa816-efa3-11e6-b9fb-22000b9a448b"           }],           "eventCreated": "2017-02-10T10:16:36.490-05:00",           "notificationDeliveryMethod": "EMAIL",           "deliveryStatus": true,           "delivered": "2017-02-10T10:17:10.100-05:00"        },              {           "eventId": 5,           "eventType": "DATA\_TRANSFER\_DOWNLOAD\_COMPLETED",           "eventPayloadEntries": [         {              "attribute": "DATA\_TRANSFER\_REQUEST\_ID",              "value": "f909e4a6-efa3-11e6-b9fb-22000b9a448b"           }],           "eventCreated": "2017-02-10T10:17:36.233-05:00",           "notificationDeliveryMethod": "EMAIL",           "deliveryStatus": true,           "delivered": "2017-02-10T10:18:10.052-05:00"   }],  "page": 1,     "limit": 100,  “totalCount": 2  } |
| Error Response | **Database Error:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  Date: Thu, 25 Aug 2016 01:03:29 GMT  Transfer-Encoding: chunked  Server: Jetty(8.1.17.v20150415)  {  "errorType": "DATABASE\_ERROR",  "message": " Failed to get notification subscriptions”  } |

## Add user query

|  |  |
| --- | --- |
| Title | Add a user query |
| Description | This API allows the user to add a new a compound query under a given name. The queries saved for the authenticated invoker and not shared with other users. |
| URL | /query/{queryName} |
| Method | PUT |
| Acceptable request representation | application/json  application/xml |
| Available response representation | application/json  application/xml |
| URL Params | queryName |
| Data Params | **JSON:**  {  "compoundQuery": {  "operator": "OR",  "queries": [  {  "attribute": "ATTR 1",  "value": "VAL 1",  "operator": "EQUAL"  },  {  "attribute": "ATTR 2",  "value": "VAL 2",  "operator": "LIKE"  }  ],  "compoundQueries": [  {  "operator": "AND",  "queries": [  {  "attribute": "ATTR 1",  "value": "VAL 1",  "operator": "EQUAL"  },  {  "attribute": "ATTR 2",  "value": "VAL 2",  "operator": "LIKE"  }  ]  },  {  "operator": "OR",  "queries": [  {  "attribute": "ATTR 3",  "value": "VAL 3",  "operator": "NOT\_EQUAL"  },  {  "attribute": "ATTR 4",  "value": "VAL 4",  "operator": "NUM\_LESS\_OR\_EQUAL"  }  ]  }  ]  },  "compoundQueryType" : "COLLECTION" |
| Success Response | HTTP/1.1 200 OK  Content-Type: application/json  HPC-API-Version: 1.0.0 |
| Error Response | **Invaid notification type:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  {  "errorType": "INVALID\_REQUEST\_INPUT",  "message": "Invalid compound metadata query",  "stackTrace": ".."  }  **Authentication Failure:**  HTTP/1.1 401 Unauthorized  Content-Type: application/json  {"errorType":"REQUEST\_AUTHENTICATION\_FAILED","message":"Access Denied: LDAP authentication failed","stackTrace":"..”}  JSON:  {  "errorType": "REQUEST\_AUTHENTICATION\_FAILED",  "message": "Access Denied: LDAP authentication failed",  "stackTrace": "…"  } |

## Update user query

|  |  |
| --- | --- |
| Title | Update a user query |
| Description | This API allows the user to update a compound query under a given name. The queries saved for the authenticated invoker and not shared with other users. |
| URL | /query/{queryName} |
| Method | POST |
| Acceptable request representation | application/json  application/xml |
| Available response representation | application/json  application/xml |
| URL Params | queryName |
| Data Params | **JSON:**  {  "compoundQuery": {  "operator": "OR",  "queries": [  {  "attribute": "ATTR 1",  "value": "VAL 1",  "operator": "EQUAL"  },  {  "attribute": "ATTR 2",  "value": "VAL 2",  "operator": "LIKE"  }  ],  "compoundQueries": [  {  "operator": "AND",  "queries": [  {  "attribute": "ATTR 1",  "value": "VAL 1",  "operator": "EQUAL"  },  {  "attribute": "ATTR 2",  "value": "VAL 2",  "operator": "LIKE"  }  ]  },  {  "operator": "OR",  "queries": [  {  "attribute": "ATTR 3",  "value": "VAL 3",  "operator": "NOT\_EQUAL"  },  {  "attribute": "ATTR 4",  "value": "VAL 4",  "operator": "NUM\_LESS\_OR\_EQUAL"  }  ]  }  ]  },  "compoundQueryType" : "DATA\_OBJECT" |
| Success Response | HTTP/1.1 200 OK  Content-Type: application/json  HPC-API-Version: 1.0.0 |
| Error Response | **Invaid notification type:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  {  "errorType": "INVALID\_REQUEST\_INPUT",  "message": "Invalid compound metadata query",  "stackTrace": ".."  }  **Authentication Failure:**  HTTP/1.1 401 Unauthorized  Content-Type: application/json  {"errorType":"REQUEST\_AUTHENTICATION\_FAILED","message":"Access Denied: LDAP authentication failed","stackTrace":"..”}  JSON:  {  "errorType": "REQUEST\_AUTHENTICATION\_FAILED",  "message": "Access Denied: LDAP authentication failed",  "stackTrace": "…"  } |

## Delete user query

|  |  |
| --- | --- |
| Title | Delete a user query |
| Description | Delete a user query |
| URL | /query/{queryName} |
| Method | DELETE |
| Acceptable request representation | application/json  application/xml |
| Available response representation | application/json  application/xml |
| URL Params | queryName |
| Data Params |  |
| Success Response | HTTP/1.1 200 OK  Content-Type: application/json  HPC-API-Version: 1.0.0 |
| Error Response | **Invaid notification type:**  **Authentication Failure:**  HTTP/1.1 401 Unauthorized  Content-Type: application/json  {"errorType":"REQUEST\_AUTHENTICATION\_FAILED","message":"Access Denied: LDAP authentication failed","stackTrace":"..”}  JSON:  {  "errorType": "REQUEST\_AUTHENTICATION\_FAILED",  "message": "Access Denied: LDAP authentication failed",  "stackTrace": "…"  } |

## Get user queries

|  |  |
| --- | --- |
| Title | Get saved queries |
| Description | Return a list of all compound metadata queries that are saved for the invoker |
| URL | /query |
| Method | GET |
| Acceptable request representation | application/json  application/xml |
| Available response representation | application/json  application/xml |
| URL Params |  |
| Data Params |  |
| Success Response | HTTP/1.1 200 OK  Access-Control-Allow-Origin: \*  Access-Control-Allow-Methods: GET, POST, DELETE, PUT  HPC-API-Version: 1.0.0  Content-Type: application/json  Date: Thu, 25 Aug 2016 01:02:02 GMT  Transfer-Encoding: chunked  Server: Jetty(8.1.17.v20150415)  {"queries": {     "name": "eran-query",     "compoundQuery":    {        "operator": "ANY",        "queries":       [                    {              "attribute": "ATTR 1",              "value": "VAL 1",              "operator": "EQUAL"           },                    {              "attribute": "ATTR 2",              "value": "VAL 2",              "operator": "LIKE"           }        ],        "compoundQueries":       [                    {              "operator": "ALL",              "queries":             [                                {                    "attribute": "ATTR 1",                    "value": "VAL 1",                    "operator": "EQUAL"                 },                                {                    "attribute": "ATTR 2",                    "value": "VAL 2",                    "operator": "LIKE"                 }              ]           },                    {              "operator": "ANY",              "queries":             [                                {                    "attribute": "ATTR 3",                    "value": "VAL 3",                    "operator": "NOT\_EQUAL"                 },                                {                    "attribute": "ATTR 4",                    "value": "VAL 4",                    "operator": "NUM\_LESS\_OR\_EQUAL"                 }              ]           }        ]     }  }} |
| Error Response | **Invaid notification user ID:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  Date: Thu, 25 Aug 2016 01:03:29 GMT  Transfer-Encoding: chunked  Server: Jetty(8.1.17.v20150415)  {  "errorType": "INVALID\_REQUEST\_INPUT",  "message": "Invalid user: <user>  } |

## Get user query

|  |  |
| --- | --- |
| Title | Get saved queries |
| Description | Return a list of all compound metadata queries that are saved for the invoker |
| URL | /query |
| Method | GET |
| Acceptable request representation | application/json  application/xml |
| Available response representation | application/json  application/xml |
| URL Params | queryName |
| Data Params |  |
| Success Response | HTTP/1.1 200 OK  Access-Control-Allow-Origin: \*  Access-Control-Allow-Methods: GET, POST, DELETE, PUT  HPC-API-Version: 1.0.0  Content-Type: application/json  Date: Thu, 25 Aug 2016 01:02:02 GMT  Transfer-Encoding: chunked  Server: Jetty(8.1.17.v20150415)  {"namedCompoundQuery": {     "name": "eran-query",     "compoundQuery":    {        "operator": "ANY",        "queries":       [                    {              "attribute": "ATTR 1",              "value": "VAL 1",              "operator": "EQUAL"           },                    {              "attribute": "ATTR 2",              "value": "VAL 2",              "operator": "LIKE"           }        ],        "compoundQueries":       [                    {              "operator": "ALL",              "queries":             [                                {                    "attribute": "ATTR 1",                    "value": "VAL 1",                    "operator": "EQUAL"                 },                                {                    "attribute": "ATTR 2",                    "value": "VAL 2",                    "operator": "LIKE"                 }              ]           },                    {              "operator": "ANY",              "queries":             [                                {                    "attribute": "ATTR 3",                    "value": "VAL 3",                    "operator": "NOT\_EQUAL"                 },                                {                    "attribute": "ATTR 4",                    "value": "VAL 4",                    "operator": "NUM\_LESS\_OR\_EQUAL"                 }              ]           }        ]     }  }} |
| Error Response | **Invaid notification user ID:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  Date: Thu, 25 Aug 2016 01:03:29 GMT  Transfer-Encoding: chunked  Server: Jetty(8.1.17.v20150415)  {  "errorType": "INVALID\_REQUEST\_INPUT",  "message": "Invalid user: <user>  } |

## 

## Get Metadata attributes

|  |  |
| --- | --- |
| Title | Get metadata attributes of both collections and data objects |
| Description | Get metadata attributes of both collections and data objects grouped by levelLabel. Each levelLabel attributes should have its own attributes, not including parent or children attributes |
| URL | /hpc-server/metadataAttributes |
| Method | GET |
| Acceptable request representation | application/json  application/xml |
| Available response representation | application/json  application/xml |
| URL Params |  |
| Data Params |  |
| Success Response | HTTP/1.1 200 OK  Date: Wed, 15 Feb 2017 18:45:02 GMT  Access-Control-Allow-Origin: \*  Access-Control-Allow-Methods: GET, POST, DELETE, PUT  HPC-API-Version: 1.0.0  Content-Type: application/json  Date: Wed, 15 Feb 2017 18:45:03 GMT  Transfer-Encoding: chunked  Server: Jetty(9.2.19.v20160908)  {      **"collectionMetadataAttributes"**:[         {            **"levelLabel"**:"PI\_Lab",          **"metadataAttributes"**:[               "collection\_type",             "contact\_name",             "name",             "pi\_name",             "registered\_by",             "configuration\_id",             "registered\_by\_name",             "uuid"          ]       },       {            **"levelLabel"**:"Project",          **"metadataAttributes"**:[               "bioinformatics\_contact",             "collection\_type",             "name",             "project\_description",             "project\_id\_CSAS",             "project\_start\_date",             "registered\_by",             "configuration\_id",             "registered\_by\_name",             "uuid"          ]       },    ],    **"dataObjectMetadataAttributes"**:[         {            **"levelLabel"**:"PI\_Lab",          **"metadataAttributes"**:[               "collection\_type",             "contact\_name",             "name",             "pi\_name",             "registered\_by",             "configuration\_id",             "registered\_by\_name",             "uuid"          ]       },       {            **"levelLabel"**:"Project",          **"metadataAttributes"**:[               "bioinformatics\_contact",             "collection\_type",             "name",             "project\_description",             "project\_id\_CSAS",             "project\_start\_date",             "registered\_by",             "configuration\_id",             "registered\_by\_name",             "uuid"          ]       },       {            **"metadataAttributes"**:[               "archive\_file\_container\_id",             "archive\_file\_id",              "configuration\_id",             "registered\_by\_name",             "software\_tool",             "source\_file\_size",             "uuid"          ]       }    ] } |
| Error Response | **Failed to get metadata attributes:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  Date: Thu, 25 Aug 2016 01:03:29 GMT  Transfer-Encoding: chunked  Server: Jetty(8.1.17.v20150415)  {  "errorType": "DATABASE\_ERROR",  "message": "Failed to get metadata attributes:”  } |

## Get Data Management Model

|  |  |
| --- | --- |
| Title | Get the data management model of a DOC |
| Description | Return the following elements of the data management model:   1. Collection metadata validation rules 2. Data Object validation rules 3. Data Hierarchy definition 4. Collection system metadata 5. Data object system metadata |
| URL | /dm/model |
| Method | GET |
| Acceptable request representation | application/json  application/xml |
| Available response representation | application/json  application/xml |
| URL Params | doc |
| Data Params |  |
| Success Response | HTTP/1.1 200 OK  Date: Fri, 16 Jun 2017 20:41:44 GMT  Access-Control-Allow-Origin: \*  Access-Control-Allow-Methods: GET, POST, DELETE, PUT  HPC-API-Version: 1.0.0  Content-Type: application/json  Date: Fri, 16 Jun 2017 20:41:48 GMT  Transfer-Encoding: chunked  Server: Jetty(9.2.19.v20160908)  {  "basePath": "/FNL\_SF\_Archive",  "collectionMetadataValidationRules": [  {  "attribute": "collection\_type",  "mandatory": true,  "validValues": [  "Project",  "PI\_Lab",  "Flowcell",  "Sample",  "Folder"  ],  "ruleEnabled": true  },  {  "attribute": "pi\_name",  "mandatory": true,  "collectionTypes": ["PI\_Lab"],  "ruleEnabled": true  },  {  "attribute": "project\_id\_CSAS",  "mandatory": true,  "collectionTypes": ["Project"],  "ruleEnabled": true  },  {  "attribute": "project\_name",  "mandatory": false,  "collectionTypes": ["Project"],  "ruleEnabled": true  },  {  "attribute": "lab\_contact",  "mandatory": false,  "collectionTypes": ["Project"],  "ruleEnabled": true  },  {  "attribute": "bioinformatics\_contact",  "mandatory": false,  "collectionTypes": ["Project"],  "ruleEnabled": true  },  {  "attribute": "project\_start\_date",  "mandatory": false,  "defaultValue": "System-Date",  "collectionTypes": ["Project"],  "ruleEnabled": true  },  {  "attribute": "grant\_funding\_agent",  "mandatory": false,  "collectionTypes": ["Project"],  "ruleEnabled": true  },  {  "attribute": "flowcell\_id",  "mandatory": true,  "collectionTypes": ["Flowcell"],  "ruleEnabled": true  },  {  "attribute": "run\_name",  "mandatory": true,  "collectionTypes": ["Flowcell"],  "ruleEnabled": true  },  {  "attribute": "run\_date",  "mandatory": true,  "collectionTypes": ["Flowcell"],  "ruleEnabled": true  },  {  "attribute": "sequencing\_platform",  "mandatory": true,  "collectionTypes": ["Flowcell"],  "ruleEnabled": true  },  {  "attribute": "sequencing\_application\_type",  "mandatory": true,  "collectionTypes": ["Flowcell"],  "ruleEnabled": true  },  {  "attribute": "read\_length",  "mandatory": true,  "collectionTypes": ["Flowcell"],  "ruleEnabled": true  },  {  "attribute": "pooling",  "mandatory": true,  "collectionTypes": ["Flowcell"],  "ruleEnabled": true  },  {  "attribute": "sample\_id",  "mandatory": true,  "collectionTypes": ["Sample"],  "ruleEnabled": true  },  {  "attribute": "sample\_name",  "mandatory": true,  "collectionTypes": ["Sample"],  "ruleEnabled": true  },  {  "attribute": "initial\_sample\_concentration\_ngul",  "mandatory": false,  "collectionTypes": ["Sample"],  "ruleEnabled": true  },  {  "attribute": "initial\_sample\_volume\_ul",  "mandatory": false,  "collectionTypes": ["Sample"],  "ruleEnabled": true  },  {  "attribute": "sfqc\_sample\_concentration\_ngul",  "mandatory": false,  "collectionTypes": ["Sample"],  "ruleEnabled": true  },  {  "attribute": "sfqc\_sample\_size",  "mandatory": false,  "collectionTypes": ["Sample"],  "ruleEnabled": true  },  {  "attribute": "RIN",  "mandatory": false,  "collectionTypes": ["Sample"],  "ruleEnabled": true  },  {  "attribute": "28s18s",  "mandatory": false,  "collectionTypes": ["Sample"],  "ruleEnabled": true  },  {  "attribute": "library\_id",  "mandatory": false,  "collectionTypes": ["Sample"],  "ruleEnabled": true  },  {  "attribute": "library\_lot",  "mandatory": false,  "collectionTypes": ["Sample"],  "ruleEnabled": true  },  {  "attribute": "library\_name",  "mandatory": false,  "collectionTypes": ["Sample"],  "ruleEnabled": true  },  {  "attribute": "sfqc\_library\_concentration\_nM",  "mandatory": false,  "collectionTypes": ["Sample"],  "ruleEnabled": true  },  {  "attribute": "sfqc\_library\_size",  "mandatory": false,  "collectionTypes": ["Sample"],  "ruleEnabled": true  },  {  "attribute": "source\_id",  "mandatory": false,  "collectionTypes": ["Sample"],  "ruleEnabled": true  },  {  "attribute": "source\_name",  "mandatory": false,  "collectionTypes": ["Sample"],  "ruleEnabled": true  },  {  "attribute": "source\_organism",  "mandatory": true,  "collectionTypes": ["Sample"],  "ruleEnabled": true  },  {  "attribute": "source\_provider",  "mandatory": false,  "collectionTypes": ["Sample"],  "ruleEnabled": true  }  ],  "dataObjectMetadataValidationRules": [  {  "attribute": "object\_name",  "mandatory": true,  "ruleEnabled": true  },  {  "attribute": "file\_type",  "mandatory": false,  "ruleEnabled": true  },  {  "attribute": "reference\_genome",  "mandatory": false,  "ruleEnabled": true  },  {  "attribute": "reference\_annotation",  "mandatory": false,  "ruleEnabled": true  },  {  "attribute": "software\_tool",  "mandatory": false,  "ruleEnabled": true  },  {  "attribute": "md5\_checksum",  "mandatory": false,  "ruleEnabled": true  },  {  "attribute": "phi\_content",  "mandatory": true,  "validValues": [  "Unspecified",  "PHI Present",  "PHI Not Present",  "Not Specified"  ],  "defaultValue": "Unspecified",  "ruleEnabled": true  },  {  "attribute": "pii\_content",  "mandatory": true,  "validValues": [  "Unspecified",  "PII Present",  "PII Not Present",  "Not Specified"  ],  "defaultValue": "Unspecified",  "ruleEnabled": true  },  {  "attribute": "data\_encryption\_status",  "mandatory": true,  "validValues": [  "Unspecified",  "Encrypted",  "Not Encrypted",  "Not Specified"  ],  "defaultValue": "Unspecified",  "ruleEnabled": true  },  {  "attribute": "data\_compression\_status",  "mandatory": true,  "validValues": [  "Unspecified",  "Compressed",  "Not Compressed",  "Not Specified"  ],  "defaultValue": "Unspecified",  "ruleEnabled": true  }  ],  "dataHierarchy": {  "collectionType": "PI\_Lab",  "isDataObjectContainer": false,  "subCollectionsHierarchies": [ {  "collectionType": "Project",  "isDataObjectContainer": false,  "subCollectionsHierarchies": [ {  "collectionType": "Flowcell",  "isDataObjectContainer": false,  "subCollectionsHierarchies": [ {  "collectionType": "Sample",  "isDataObjectContainer": true  }]  }]  }]  },  "collectionSystemGeneratedMetadataAttributeNames": [  "uuid",  "registered\_by",  "registered\_by\_name",  "configuration\_id",  "metadata\_updated"  ],  "dataObjectSystemGeneratedMetadataAttributeNames": [  "uuid",  "registered\_by",  "registered\_by\_name",  "configuration\_id",  "source\_file\_container\_id",  "source\_file\_id",  "archive\_file\_container\_id",  "archive\_file\_id",  "data\_transfer\_request\_id",  "data\_transfer\_status",  "data\_transfer\_type",  "data\_transfer\_started",  "data\_transfer\_completed",  "source\_file\_size",  "archive\_caller\_object\_id",  "checksum",  "metadata\_updated"  ]  } |
| Error Response | **Invalid DOC:**  HTTP/1.1 400 Bad Request  Date: Fri, 16 Jun 2017 20:49:31 GMT  Content-Type: application/json  HPC-API-Version: 1.0.0  Date: Fri, 16 Jun 2017 20:49:33 GMT  Transfer-Encoding: chunked  Server: Jetty(9.2.19.v20160908)  {  "errorType": "INVALID\_REQUEST\_INPUT",  "message": "DOC not supported: invalid-doc-name"  } |

## Add Bookmark

|  |  |
| --- | --- |
| Title | Add a bookmark |
| Description | This API allows the user to add a new bookmark. It also allows the system admin or group admin to add bookmarks for other users. |
| URL | /bookmark/{bookmark} |
| Method | PUT |
| Acceptable request representation | application/json  application/xml |
| Available response representation | application/json  application/xml |
| URL Params | bookmarkName |
| Data Params | **JSON:**  {  "path" : "/Path/To/Bookmark",  "group" : "A bookmark group name",  "userId" : "The user Id of the NIH AD account of the user",  "permission": "The permission to set for the path"  }  **Important note**: The fields 'userId' and 'permission' are available only for system admin. and group admin. to set bookmark and permission for other users. If no permission is specified, then the existing permission is left as is. |
| Sample Data Params | {  "path" : "/TEST\_NO\_HIER\_Archive/test\_collection",  "group" : "test\_group",  "userId" : "menons2",  "permission": "WRITE"  } |
| Success Response | HTTP/1.1 201 Created  Content-Type: application/json  HPC-API-Version: 1.0.0 |
| Error Response | **Invaid bookmark:**  In case mandatory bookmark path is not provided.  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  {  "errorType": "INVALID\_REQUEST\_INPUT",  "message": "Invalid bookmark request: Null or empty bookmark path",  **}**  **Bookmark already exists:**  In case bookmark exists  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  {  "errorType": "INVALID\_REQUEST\_INPUT",  "message": "Bookmark name already exists: <bookmark-name>",  **}**  **Authentication Failure:**  HTTP/1.1 401 Unauthorized  Content-Type: application/json  {"errorType":"REQUEST\_AUTHENTICATION\_FAILED","message":"Access Denied: LDAP authentication failed","stackTrace":"..”}  JSON:  {  "errorType": "REQUEST\_AUTHENTICATION\_FAILED",  "message": "Access Denied: LDAP authentication failed",  "stackTrace": "…"  } |

## Update Bookmark

|  |  |
| --- | --- |
| Title | Update a bookmark |
| Description | This API allows the user to update a bookmark’s path and/or group. It also allows the system admin or group admin to update bookmark's path, group and permissions for other users. |
| URL | /bookmark/{bookmarkName} |
| Method | POST |
| Acceptable request representation | application/json  application/xml |
| Available response representation | application/json  application/xml |
| URL Params | bookmarkName |
| Data Params | **JSON:**  {  "path" : "/Updated/Path/To/Bookmark",  "group" : "A bookmark updated group name",  "userId" : "The user Id of the NIH AD account of the user",  "permission": "The permission to set for the path"  }  **Important note**: The fields 'userId' and 'permission' are available only for system admin. and group admin. to set bookmark and permission for other users. |
| Sample Data Params | {  "path" : "/TEST\_NO\_HIER\_Archive/test\_collection",  "group" : "test\_group",  "userId" : "menons2",  "permission": "WRITE"  } |
| Success Response | HTTP/1.1 200 OK  Content-Type: application/json  HPC-API-Version: 1.0.0 |
| Error Response | **Bookmark doesn’t exist:**  In case bookmark doesn’t exist  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  {  "errorType": "INVALID\_REQUEST\_INPUT",  "message": "Bookmark name doesn’t exist: <bookmark-name>",  **}**  **Authentication Failure:**  HTTP/1.1 401 Unauthorized  Content-Type: application/json  {"errorType":"REQUEST\_AUTHENTICATION\_FAILED","message":"Access Denied: LDAP authentication failed","stackTrace":"..”}  JSON:  {  "errorType": "REQUEST\_AUTHENTICATION\_FAILED",  "message": "Access Denied: LDAP authentication failed",  "stackTrace": "…"  } |

## Delete Bookmark

|  |  |
| --- | --- |
| Title | Delete a bookmark |
| Description | Delete a bookmark |
| URL | /bookmark/{bookmarkName} |
| Method | DELETE |
| Acceptable request representation | application/json  application/xml |
| Available response representation | application/json  application/xml |
| URL Params | bookmarkName |
| Data Params |  |
| Success Response | HTTP/1.1 200 OK  Content-Type: application/json  HPC-API-Version: 1.0.0 |
| Error Response | **Bookmark doesn’t exist:**  In case bookmark doesn’t exist  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  {  "errorType": "INVALID\_REQUEST\_INPUT",  "message": "Bookmark name doesn’t exist: <bookmark-name>",  **}**  **Authentication Failure:**  HTTP/1.1 401 Unauthorized  Content-Type: application/json  {"errorType":"REQUEST\_AUTHENTICATION\_FAILED","message":"Access Denied: LDAP authentication failed","stackTrace":"..”}  JSON:  {  "errorType": "REQUEST\_AUTHENTICATION\_FAILED",  "message": "Access Denied: LDAP authentication failed",  "stackTrace": "…"  } |

## Get Bookmarks

|  |  |
| --- | --- |
| Title | Get all saved bookmarks |
| Description | Return a list of all bookmarks that are saved for the invoker |
| URL | /bookmark |
| Method | GET |
| Acceptable request representation | application/json  application/xml |
| Available response representation | application/json  application/xml |
| URL Params |  |
| Data Params |  |
| Success Response | HTTP/1.1 200 OK  Access-Control-Allow-Origin: \*  Access-Control-Allow-Methods: GET, POST, DELETE, PUT  HPC-API-Version: 1.0.0  Content-Type: application/json  Date: Thu, 25 Aug 2016 01:02:02 GMT  Transfer-Encoding: chunked  Server: Jetty(8.1.17.v20150415)  {"bookmarks": [{  "name": "bookmark-name",  "group": "my-group",  "path": "/Bookmark/Path",  "created": "2017-06-21T09:43:43.641-04:00",  "updated": "2017-06-21T09:58:27.976-04:00"  }]} |
| Error Response | **Authentication Failure:**  HTTP/1.1 401 Unauthorized  Content-Type: application/json  {"errorType":"REQUEST\_AUTHENTICATION\_FAILED","message":"Access Denied: LDAP authentication failed","stackTrace":"..”}  JSON:  {  "errorType": "REQUEST\_AUTHENTICATION\_FAILED",  "message": "Access Denied: LDAP authentication failed",  "stackTrace": "…"  } |

## Get Bookmark

|  |  |
| --- | --- |
| Title | Get a bookmark |
| Description | Return a bookmark by name |
| URL | /bookmark/{bookmarkName} |
| Method | GET |
| Acceptable request representation | application/json  application/xml |
| Available response representation | application/json  application/xml |
| URL Params | bookmarkName |
| Data Params |  |
| Success Response | HTTP/1.1 200 OK  Access-Control-Allow-Origin: \*  Access-Control-Allow-Methods: GET, POST, DELETE, PUT  HPC-API-Version: 1.0.0  Content-Type: application/json  Date: Thu, 25 Aug 2016 01:02:02 GMT  Transfer-Encoding: chunked  Server: Jetty(8.1.17.v20150415)  {"bookmark": {  "name": "bookmark-name",  "group": "my-group",  "path": "/Bookmark/Path",  "created": "2017-06-21T09:43:43.641-04:00",  "updated": "2017-06-21T09:58:27.976-04:00"  }} |
| Error Response | **Authentication Failure:**  HTTP/1.1 401 Unauthorized  Content-Type: application/json  {"errorType":"REQUEST\_AUTHENTICATION\_FAILED","message":"Access Denied: LDAP authentication failed","stackTrace":"..”}  JSON:  {  "errorType": "REQUEST\_AUTHENTICATION\_FAILED",  "message": "Access Denied: LDAP authentication failed",  "stackTrace": "…"  } |

## Generate Report

|  |  |
| --- | --- |
| Title | Generate usage summary report |
| Description | Authorized HPC DME users can generate the following usage summary reports:  USAGE\_SUMMARY\_REPORT:  USAGE\_SUMMARY\_BY\_WEEKLY\_REPORT:  USAGE\_SUMMARY\_BY\_DOC\_REPORT:  USAGE\_SUMMARY\_BY\_DOC\_BY\_WEEKLY\_REPORT:  USAGE\_SUMMARY\_BY\_USER\_REPORT:  USAGE\_SUMMARY\_BY\_USER\_BY\_WEEKLY\_REPORT: |
| URL | /report |
| Method | POST |
| Acceptable request representation | application/json  application/xml |
| Available response representation | application/json  application/xml |
| URL Params |  |
| Data Params | **JSON:**  {  "type": "USAGE\_SUMMARY"  }  or  {  "type": "USAGE\_SUMMARY\_BY\_DATE\_RANGE",  "fromDate": "08/01/2016",  "toDate": "08/20/2016"  }  Or  {  "type": "USAGE\_SUMMARY\_BY\_DOC",  "doc": "FNLCR"  }  Or  {  "type": "USAGE\_SUMMARY\_BY\_DOC\_BY\_DATE\_RANGE",  "doc": "FNLCR",  "fromDate": "08/01/2016",  "toDate": "08/20/2016"  }  Or  {  "type": "USAGE\_SUMMARY\_BY\_USER",  "user": "konkapv"  }  Or  {  "type": "USAGE\_SUMMARY\_BY\_USER\_BY\_DATE\_RANGE",  "user": "konkapv",  "fromDate": "08/01/2016",  "toDate": "08/20/2016"  } |
| Success Response | HTTP/1.1 200 OK  Content-Type: application/json  HPC-API-Version: 1.0.0 |
| Error Response | **Invalid Report type:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  {  "errorType": "INVALID\_REQUEST\_INPUT",  "message": "Report type is missing",  "stackTrace": ".."  }  **Invalid Date format:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  HPC-API-Version: 1.0.0  {  "errorType": "INVALID\_REQUEST\_INPUT",  "message": "Invalid fromDate format. Valid format is mm/dd/yyyy",  "stackTrace": ".."  }  **Authentication Failure:**  HTTP/1.1 401 Unauthorized  Content-Type: application/json  {"errorType":"REQUEST\_AUTHENTICATION\_FAILED","message":"Access Denied: LDAP authentication failed","stackTrace":"..”}  JSON:  {  "errorType": "REQUEST\_AUTHENTICATION\_FAILED",  "message": "Access Denied: LDAP authentication failed",  "stackTrace": "…"  } |

## Bulk Move of Collections and Data Objects

|  |  |
| --- | --- |
| Title | Bulk moving of collections and data objects |
| Description | Performing a bulk request of moving collections and/or data objects. |
| URL | /move |
| Method | POST |
| Acceptable request representation | application/json  application/xml |
| Available response representation | application/json  application/xml |
| URL Params |  |
| Data Params | **JSON:**  {  "moveRequests": [{  "sourcePath": "/FNL\_SF\_Archive/eran-pi-lab/eran-project/eran-flowcell/eran-sample/eran-data-object-bulk-1-moved",  "destinationPath": "moved-data-object"  },  {  "sourcePath": "/FNL\_SF\_Archive/eran-pi-lab/eran-project/eran-flowcell/eran-sample-2",  "destinationPath": "moved-collection"  }  ]  } |
| Success Response | HTTP/1.1 200 OK  Content-Type: application/json  HPC-API-Version: 1.0.0  {  "moveResponses": [{  "request": {  "sourcePath": "/FNL\_SF\_Archive/eran-pi-lab/eran-project/eran-flowcell/eran-sample/eran-data-object-bulk-1-moved",  "destinationPath": "moved-data-object"  },  "result": false,  "message": "Path doesn't exist"  },  {  "request": {  "sourcePath": "/FNL\_SF\_Archive/eran-pi-lab/eran-project/eran-flowcell/eran-sample-2",  "destinationPath": "moveed-collection"  },  "result": true  }  ]  } |
| Error Response | **Authentication Failure:**  HTTP/1.1 401 Unauthorized  Content-Type: application/json  {"errorType":"REQUEST\_AUTHENTICATION\_FAILED","message":"Access Denied: LDAP authentication failed","stackTrace":"..”}  JSON:  {  "errorType": "REQUEST\_AUTHENTICATION\_FAILED",  "message": "Access Denied: LDAP authentication failed",  "stackTrace": "…"  } |

# APPENDIX A: HPC SERVER API GRAMMER

|  |
| --- |
| {      **"grammars"**:{         **"xs.schema"**:[            {               **"@attributeFormDefault"**:"unqualified",             **"@elementFormDefault"**:"unqualified",             **"@targetNamespace"**:"http://hpc.nci.nih.gov/domain/report",             **"xs.complexType"**:[                  {                     **"@name"**:"HpcReport",                   **"xs.sequence"**:{                        **"xs.element"**:[                           {                              **"@name"**:"type",                            **"@type"**:"xs:string"                         },                         {                              **"@name"**:"doc",                            **"@type"**:"xs:string"                         },                         {                              **"@name"**:"user",                            **"@type"**:"xs:string"                         },                         {                              **"@name"**:"fromDate",                            **"@type"**:"xs:date"                         },                         {                              **"@name"**:"ToDate",                            **"@type"**:"xs:date"                         },                         {                              **"@name"**:"generatedOn",                            **"@type"**:"xs:date"                         },                         {                              **"@maxOccurs"**:"unbounded",                            **"@minOccurs"**:"0",                            **"@name"**:"reportEntries",                            **"@nillable"**:"true",                            **"@type"**:"tns:HpcReportEntry"                         }                      ]                   }                },                {                     **"@name"**:"HpcReportEntry",                   **"xs.sequence"**:{                        **"xs.element"**:[                           {                              **"@name"**:"attribute",                            **"@type"**:"xs:string"                         },                         {                              **"@name"**:"value",                            **"@type"**:"xs:string"                         }                      ]                   }                }             ],             **"xs.simpleType"**:[                  {                     **"@name"**:"HpcReportType",                   **"xs.restriction"**:{                        **"@base"**:"xs:string",                      **"xs.enumeration"**:[                           {                              **"@value"**:"USAGE\_SUMMARY"                         },                         {                              **"@value"**:"USAGE\_SUMMARY\_BY\_DATE\_RANGE"                         },                         {                              **"@value"**:"USAGE\_SUMMARY\_BY\_DOC"                         },                         {                              **"@value"**:"USAGE\_SUMMARY\_BY\_DOC\_BY\_DATE\_RANGE"                         },                         {                              **"@value"**:"USAGE\_SUMMARY\_BY\_USER"                         },                         {                              **"@value"**:"USAGE\_SUMMARY\_BY\_USER\_BY\_DATE\_RANGE"                         }                      ]                   }                },                {                     **"@name"**:"HpcReportEntryAttribute",                   **"xs.restriction"**:{                        **"@base"**:"xs:string",                      **"xs.enumeration"**:[                           {                              **"@value"**:"REPORT\_GENERATED\_ON"                         },                         {                              **"@value"**:"FROM\_DATE"                         },                         {                              **"@value"**:"TO\_DATE"                         },                         {                              **"@value"**:"TYPE"                         },                         {                              **"@value"**:"DOC"                         },                         {                              **"@value"**:"USER\_ID"                         },                         {                              **"@value"**:"TOTAL\_DATA\_SIZE"                         },                         {                              **"@value"**:"TOTAL\_NUM\_OF\_REGISTERED\_USERS"                         },                         {                              **"@value"**:"TOTAL\_NUM\_OF\_DATA\_OBJECTS"                         },                         {                              **"@value"**:"TOTAL\_NUM\_OF\_COLLECTIONS"                         },                         {                              **"@value"**:"TOTAL\_NUMBER\_OF\_META\_ATTRS"                         },                         {                              **"@value"**:"LARGEST\_FILE\_SIZE"                         },                         {                              **"@value"**:"AVERAGE\_FILE\_SIZE"                         },                         {                              **"@value"**:"SMALLEST\_FILE\_SIZE"                         },                         {                              **"@value"**:"FILE\_SIZE\_BELOW\_1MB"                         },                         {                              **"@value"**:"FILE\_SIZE\_1MB\_10MB"                         },                         {                              **"@value"**:"FILE\_SIZE\_10MB\_50MB"                         },                         {                              **"@value"**:"FILE\_SIZE\_50MB\_100MB"                         },                         {                              **"@value"**:"FILE\_SIZE\_100MB\_500MB"                         },                         {                              **"@value"**:"FILE\_SIZE\_500MB\_1GB"                         },                         {                              **"@value"**:"FILE\_SIZE\_1GB\_10GB"                         },                         {                              **"@value"**:"FILE\_SIZE\_OVER\_10GB"                         }                      ]                   }                }             ]          },          {               **"@attributeFormDefault"**:"unqualified",             **"@elementFormDefault"**:"unqualified",             **"@targetNamespace"**:"http://hpc.nci.nih.gov/domain/datatransfer",             **"xs.complexType"**:{                  **"@name"**:"HpcFileLocation",                **"xs.sequence"**:{                     **"xs.element"**:[                        {                           **"@name"**:"fileContainerId",                         **"@type"**:"xs:string"                      },                      {                           **"@name"**:"fileId",                         **"@type"**:"xs:string"                      }                   ]                }             },             **"xs.simpleType"**:{                  **"@name"**:"HpcDataTransferType",                **"xs.restriction"**:{                     **"@base"**:"xs:string",                   **"xs.enumeration"**:[                        {                           **"@value"**:"GLOBUS"                      },                      {                           **"@value"**:"S3"                      }                   ]                }             }          },          {               **"@attributeFormDefault"**:"unqualified",             **"@elementFormDefault"**:"unqualified",             **"@targetNamespace"**:"http://hpc.nci.nih.gov/dto/report",             **"xs.import"**:{                  **"@namespace"**:"http://hpc.nci.nih.gov/domain/report"             },             **"xs.element"**:[                  {                     **"@name"**:"HpcReportDTO",                   **"xs.complexType"**:{                        **"xs.sequence"**:{                           **"xs.element"**:[                              {                                 **"@name"**:"type",                               **"@type"**:"xs:string"                            },                            {                                 **"@name"**:"doc",                               **"@type"**:"xs:string"                            },                            {                                 **"@name"**:"user",                               **"@type"**:"xs:string"                            },                            {                                 **"@name"**:"fromDate",                               **"@type"**:"xs:date"                            },                            {                                 **"@name"**:"ToDate",                               **"@type"**:"xs:date"                            },                            {                                 **"@name"**:"generatedOn",                               **"@type"**:"xs:date"                            },                            {                                 **"@maxOccurs"**:"unbounded",                               **"@minOccurs"**:"0",                               **"@name"**:"reportEntries",                               **"@nillable"**:"true",                               **"@type"**:"ns1:HpcReportEntry"                            }                         ]                      }                   }                },                {                     **"@name"**:"HpcReportRequestDTO",                   **"xs.complexType"**:{                        **"xs.sequence"**:{                           **"xs.element"**:[                              {                                 **"@name"**:"type",                               **"@type"**:"xs:string"                            },                            {                                 **"@maxOccurs"**:"unbounded",                               **"@name"**:"doc",                               **"@type"**:"xs:string"                            },                            {                                 **"@maxOccurs"**:"unbounded",                               **"@name"**:"user",                               **"@type"**:"xs:string"                            },                            {                                 **"@name"**:"fromDate",                               **"@type"**:"xs:string"                            },                            {                                 **"@name"**:"toDate",                               **"@type"**:"xs:string"                            }                         ]                      }                   }                },                {                     **"@name"**:"HpcReportsDTO",                   **"xs.complexType"**:{                        **"xs.sequence"**:{                           **"xs.element"**:{                              **"@maxOccurs"**:"unbounded",                            **"@minOccurs"**:"0",                            **"@name"**:"reports",                            **"@nillable"**:"true",                            **"@type"**:"ns1:HpcReport"                         }                      }                   }                }             ]          },          {               **"@attributeFormDefault"**:"unqualified",             **"@elementFormDefault"**:"unqualified",             **"@targetNamespace"**:"http://hpc.nci.nih.gov/dto/security",             **"xs.import"**:[                  {                     **"@namespace"**:"http://hpc.nci.nih.gov/domain/user"                },                {                     **"@namespace"**:"http://hpc.nci.nih.gov/domain/datatransfer"                }             ],             **"xs.element"**:[                  {                     **"@name"**:"HpcAuthenticationResponseDTO",                   **"xs.complexType"**:{                        **"xs.sequence"**:{                           **"xs.element"**:[                              {                                 **"@name"**:"authenticated",                               **"@type"**:"xs:boolean"                            },                            {                                 **"@name"**:"userRole",                               **"@type"**:"xs:string"                            },                            {                                 **"@name"**:"token",                               **"@type"**:"xs:string"                            }                         ]                      }                   }                },                {                     **"@name"**:"HpcGroupRequestDTO",                   **"xs.complexType"**:{                        **"xs.sequence"**:{                           **"xs.element"**:[                              {                                 **"@name"**:"group",                               **"@type"**:"xs:string"                            },                            {                                 **"@maxOccurs"**:"unbounded",                               **"@minOccurs"**:"0",                               **"@name"**:"addUserIds",                               **"@nillable"**:"true",                               **"@type"**:"xs:string"                            },                            {                                 **"@maxOccurs"**:"unbounded",                               **"@minOccurs"**:"0",                               **"@name"**:"deleteUserIds",                               **"@nillable"**:"true",                               **"@type"**:"xs:string"                            }                         ]                      }                   }                },                {                     **"@name"**:"HpcGroupResponseDTO",                   **"xs.complexType"**:{                        **"xs.sequence"**:{                           **"xs.element"**:[                              {                                 **"@name"**:"group",                               **"@type"**:"xs:string"                            },                            {                                 **"@name"**:"result",                               **"@type"**:"xs:boolean"                            },                            {                                 **"@name"**:"message",                               **"@type"**:"xs:string"                            },                            {                                 **"@maxOccurs"**:"unbounded",                               **"@minOccurs"**:"0",                               **"@name"**:"userGroupResponses",                               **"@nillable"**:"true",                               **"@type"**:"tns:HpcUserGroupResponseDTO"                            }                         ]                      }                   }                },                {                     **"@name"**:"HpcSystemAccountDTO",                   **"xs.complexType"**:{                        **"xs.sequence"**:{                           **"xs.element"**:[                              {                                 **"@name"**:"account",                               **"@type"**:"ns1:HpcIntegratedSystemAccount"                            },                            {                                 **"@name"**:"dataTransferType",                               **"@type"**:"xs:string"                            }                         ]                      }                   }                },                {                     **"@name"**:"HpcUpdateUserRequestDTO",                   **"xs.complexType"**:{                        **"xs.sequence"**:{                           **"xs.element"**:[                              {                                 **"@name"**:"firstName",                               **"@type"**:"xs:string"                            },                            {                                 **"@name"**:"lastName",                               **"@type"**:"xs:string"                            },                            {                                 **"@name"**:"doc",                               **"@type"**:"xs:string"                            },                            {                                 **"@name"**:"userRole",                               **"@type"**:"xs:string"                            }                         ]                      }                   }                },                {                     **"@name"**:"HpcUserDTO",                   **"xs.complexType"**:{                        **"xs.sequence"**:{                           **"xs.element"**:[                              {                                 **"@name"**:"nciAccount",                               **"@type"**:"ns1:HpcNciAccount"                            },                            {                                 **"@name"**:"dataManagementAccount",                               **"@type"**:"ns1:HpcIntegratedSystemAccount"                            },                            {                                 **"@name"**:"userRole",                               **"@type"**:"xs:string"                            }                         ]                      }                   }                }             ],             **"xs.complexType"**:{                  **"@name"**:"HpcUserGroupResponseDTO",                **"xs.sequence"**:{                     **"xs.element"**:[                        {                           **"@name"**:"userId",                         **"@type"**:"xs:string"                      },                      {                           **"@name"**:"result",                         **"@type"**:"xs:boolean"                      },                      {                           **"@name"**:"message",                         **"@type"**:"xs:string"                      }                   ]                }             }          },          {               **"@attributeFormDefault"**:"unqualified",             **"@elementFormDefault"**:"unqualified",             **"@targetNamespace"**:"http://hpc.nci.nih.gov/dto/notification",             **"xs.import"**:{                  **"@namespace"**:"http://hpc.nci.nih.gov/domain/notification"             },             **"xs.element"**:[                  {                     **"@name"**:"HpcNotificationSubscriptionListDTO",                   **"xs.complexType"**:{                        **"xs.sequence"**:{                           **"xs.element"**:{                              **"@maxOccurs"**:"unbounded",                            **"@minOccurs"**:"0",                            **"@name"**:"subscriptions",                            **"@nillable"**:"true",                            **"@type"**:"ns1:HpcNotificationSubscription"                         }                      }                   }                },                {                     **"@name"**:"HpcNotificationSubscriptionsRequestDTO",                   **"xs.complexType"**:{                        **"xs.sequence"**:{                           **"xs.element"**:[                              {                                 **"@maxOccurs"**:"unbounded",                               **"@minOccurs"**:"0",                               **"@name"**:"addUpdateSubscriptions",                               **"@nillable"**:"true",                               **"@type"**:"ns1:HpcNotificationSubscription"                            },                            {                                 **"@maxOccurs"**:"unbounded",                               **"@minOccurs"**:"0",                               **"@name"**:"deleteSubscriptions",                               **"@nillable"**:"true",                               **"@type"**:"xs:string"                            }                         ]                      }                   }                }             ]          },          {               **"@attributeFormDefault"**:"unqualified",             **"@elementFormDefault"**:"unqualified",             **"@targetNamespace"**:"http://hpc.nci.nih.gov/domain/metadata",             **"xs.complexType"**:[                  {                     **"@name"**:"HpcMetadataEntry",                   **"xs.sequence"**:{                        **"xs.element"**:[                           {                              **"@name"**:"attribute",                            **"@type"**:"xs:string"                         },                         {                              **"@name"**:"value",                            **"@type"**:"xs:string"                         },                         {                              **"@name"**:"unit",                            **"@type"**:"xs:string"                         },                         {                              **"@minOccurs"**:"0",                            **"@name"**:"level",                            **"@type"**:"xs:int"                         }                      ]                   }                },                {                     **"@name"**:"HpcMetadataEntries",                   **"xs.sequence"**:{                        **"xs.element"**:[                           {                              **"@maxOccurs"**:"unbounded",                            **"@name"**:"selfMetadataEntries",                            **"@type"**:"tns:HpcMetadataEntry"                         },                         {                              **"@maxOccurs"**:"unbounded",                            **"@name"**:"parentMetadataEntries",                            **"@type"**:"tns:HpcMetadataEntry"                         }                      ]                   }                },                {                     **"@name"**:"HpcCompoundMetadataQuery",                   **"xs.sequence"**:{                        **"xs.element"**:[                           {                              **"@name"**:"operator",                            **"@type"**:"xs:string"                         },                         {                              **"@maxOccurs"**:"unbounded",                            **"@minOccurs"**:"0",                            **"@name"**:"queries",                            **"@nillable"**:"true",                            **"@type"**:"tns:HpcMetadataQuery"                         },                         {                              **"@maxOccurs"**:"unbounded",                            **"@minOccurs"**:"0",                            **"@name"**:"compoundQueries",                            **"@nillable"**:"true",                            **"@type"**:"tns:HpcCompoundMetadataQuery"                         }                      ]                   }                },                {                     **"@name"**:"HpcMetadataQuery",                   **"xs.sequence"**:{                        **"xs.element"**:[                           {                              **"@name"**:"attribute",                            **"@type"**:"xs:string"                         },                         {                              **"@name"**:"value",                            **"@type"**:"xs:string"                         },                         {                              **"@name"**:"operator",                            **"@type"**:"xs:string"                         },                         {                              **"@name"**:"levelFilter",                            **"@type"**:"tns:HpcMetadataQueryLevelFilter"                         }                      ]                   }                },                {                     **"@name"**:"HpcMetadataQueryLevelFilter",                   **"xs.sequence"**:{                        **"xs.element"**:[                           {                              **"@name"**:"level",                            **"@type"**:"xs:int"                         },                         {                              **"@name"**:"operator",                            **"@type"**:"xs:string"                         }                      ]                   }                },                {                     **"@name"**:"HpcNamedCompoundMetadataQuery",                   **"xs.sequence"**:{                        **"xs.element"**:[                           {                              **"@name"**:"name",                            **"@type"**:"xs:string"                         },                         {                              **"@name"**:"compoundQuery",                            **"@type"**:"tns:HpcCompoundMetadataQuery"                         }                      ]                   }                },                {                     **"@name"**:"HpcMetadataValidationRule",                   **"xs.sequence"**:{                        **"xs.element"**:[                           {                              **"@name"**:"attribute",                            **"@type"**:"xs:string"                         },                         {                              **"@name"**:"mandatory",                            **"@type"**:"xs:boolean"                         },                         {                              **"@maxOccurs"**:"unbounded",                            **"@minOccurs"**:"0",                            **"@name"**:"validValues",                            **"@nillable"**:"true",                            **"@type"**:"xs:string"                         },                         {                              **"@name"**:"defaultValue",                            **"@type"**:"xs:string"                         },                         {                              **"@name"**:"defaultUnit",                            **"@type"**:"xs:string"                         },                         {                              **"@maxOccurs"**:"unbounded",                            **"@minOccurs"**:"0",                            **"@name"**:"collectionTypes",                            **"@nillable"**:"true",                            **"@type"**:"xs:string"                         },                         {                              **"@name"**:"ruleEnabled",                            **"@type"**:"xs:boolean"                         },                         {                              **"@maxOccurs"**:"unbounded",                            **"@minOccurs"**:"0",                            **"@name"**:"doc",                            **"@nillable"**:"true",                            **"@type"**:"xs:string"                         }                      ]                   }                },                {                     **"@name"**:"HpcMetadataLevelAttributes",                   **"xs.sequence"**:{                        **"xs.element"**:[                           {                              **"@minOccurs"**:"0",                            **"@name"**:"level",                            **"@type"**:"xs:int"                         },                         {                              **"@maxOccurs"**:"unbounded",                            **"@minOccurs"**:"0",                            **"@name"**:"metadataAttributes",                            **"@nillable"**:"true",                            **"@type"**:"xs:string"                         }                      ]                   }                }             ],             **"xs.simpleType"**:[                  {                     **"@name"**:"HpcCompoundMetadataQueryOperator",                   **"xs.restriction"**:{                        **"@base"**:"xs:string",                      **"xs.enumeration"**:[                           {                              **"@value"**:"OR"                         },                         {                              **"@value"**:"AND"                         }                      ]                   }                },                {                     **"@name"**:"HpcMetadataQueryOperator",                   **"xs.restriction"**:{                        **"@base"**:"xs:string",                      **"xs.enumeration"**:[                           {                              **"@value"**:"EQUAL"                         },                         {                              **"@value"**:"NOT\_EQUAL"                         },                         {                              **"@value"**:"LIKE"                         },                         {                              **"@value"**:"NUM\_LESS\_THAN"                         },                         {                              **"@value"**:"NUM\_GREATER\_THAN"                         },                         {                              **"@value"**:"NUM\_LESS\_OR\_EQUAL"                         },                         {                              **"@value"**:"NUM\_GREATER\_OR\_EQUAL"                         }                      ]                   }                }             ],             **"xs.element"**:[                  {                     **"@name"**:"HpcMetadataEntry",                   **"@type"**:"tns:HpcMetadataEntry"                },                {                     **"@name"**:"HpcMetadataEntrys",                   **"xs.complexType"**:{                        **"xs.sequence"**:{                           **"xs.element"**:{                              **"@maxOccurs"**:"unbounded",                            **"@minOccurs"**:"0",                            **"@ref"**:"tns:HpcMetadataEntry"                         }                      }                   }                }             ]          },          {               **"@attributeFormDefault"**:"unqualified",             **"@elementFormDefault"**:"unqualified",             **"@targetNamespace"**:"http://hpc.nci.nih.gov/domain/user",             **"xs.complexType"**:[                  {                     **"@name"**:"HpcNciAccount",                   **"xs.sequence"**:{                        **"xs.element"**:[                           {                              **"@name"**:"userId",                            **"@type"**:"xs:string"                         },                         {                              **"@name"**:"firstName",                            **"@type"**:"xs:string"                         },                         {                              **"@name"**:"lastName",                            **"@type"**:"xs:string"                         },                         {                              **"@name"**:"doc",                            **"@type"**:"xs:string"                         }                      ]                   }                },                {                     **"@name"**:"HpcIntegratedSystemAccount",                   **"xs.sequence"**:{                        **"xs.element"**:[                           {                              **"@name"**:"username",                            **"@type"**:"xs:string"                         },                         {                              **"@name"**:"password",                            **"@type"**:"xs:string"                         },                         {                              **"@name"**:"integratedSystem",                            **"@type"**:"xs:string"                         }                      ]                   }                },                {                     **"@name"**:"HpcGroupResponse",                   **"xs.sequence"**:{                        **"xs.element"**:[                           {                              **"@name"**:"group",                            **"@type"**:"xs:string"                         },                         {                              **"@name"**:"result",                            **"@type"**:"xs:boolean"                         },                         {                              **"@name"**:"message",                            **"@type"**:"xs:string"                         },                         {                              **"@maxOccurs"**:"unbounded",                            **"@minOccurs"**:"0",                            **"@name"**:"groupuser",                            **"@nillable"**:"true",                            **"@type"**:"tns:HpcGroupUserResponse"                         }                      ]                   }                },                {                     **"@name"**:"HpcGroupUserResponse",                   **"xs.sequence"**:{                        **"xs.element"**:[                           {                              **"@name"**:"userId",                            **"@type"**:"xs:string"                         },                         {                              **"@name"**:"result",                            **"@type"**:"xs:boolean"                         },                         {                              **"@name"**:"message",                            **"@type"**:"xs:string"                         }                      ]                   }                }             ],             **"xs.simpleType"**:[                  {                     **"@name"**:"HpcIntegratedSystem",                   **"xs.restriction"**:{                        **"@base"**:"xs:string",                      **"xs.enumeration"**:[                           {                              **"@value"**:"GLOBUS"                         },                         {                              **"@value"**:"IRODS"                         },                         {                              **"@value"**:"CLEVERSAFE"                         }                      ]                   }                },                {                     **"@name"**:"HpcUserRole",                   **"xs.restriction"**:{                        **"@base"**:"xs:string",                      **"xs.enumeration"**:[                           {                              **"@value"**:"USER"                         },                         {                              **"@value"**:"GROUP\_ADMIN"                         },                         {                              **"@value"**:"SYSTEM\_ADMIN"                         },                         {                              **"@value"**:"NOT\_REGISTERED"                         }                      ]                   }                }             ]          },          {               **"@attributeFormDefault"**:"unqualified",             **"@elementFormDefault"**:"unqualified",             **"@targetNamespace"**:"http://hpc.nci.nih.gov/dto/datamanagement",             **"xs.import"**:[                  {                     **"@namespace"**:"http://hpc.nci.nih.gov/domain/datamanagement"                },                {                     **"@namespace"**:"http://hpc.nci.nih.gov/domain/datatransfer"                },                {                     **"@namespace"**:"http://hpc.nci.nih.gov/domain/metadata"                }             ],             **"xs.element"**:[                  {                     **"@name"**:"HpcCollectionListDTO",                   **"xs.complexType"**:{                        **"xs.sequence"**:{                           **"xs.element"**:[                              {                                 **"@maxOccurs"**:"unbounded",                               **"@minOccurs"**:"0",                               **"@name"**:"collections",                               **"@nillable"**:"true",                               **"@type"**:"tns:HpcCollectionDTO"                            },                            {                                 **"@maxOccurs"**:"unbounded",                               **"@minOccurs"**:"0",                               **"@name"**:"collectionPaths",                               **"@nillable"**:"true",                               **"@type"**:"xs:string"                            },                            {                                 **"@minOccurs"**:"0",                               **"@name"**:"page",                               **"@type"**:"xs:int"                            },                            {                                 **"@minOccurs"**:"0",                               **"@name"**:"limit",                               **"@type"**:"xs:int"                            }                         ]                      }                   }                },                {                     **"@name"**:"HpcCompoundMetadataQueryDTO",                   **"xs.complexType"**:{                        **"xs.sequence"**:{                           **"xs.element"**:[                              {                                 **"@name"**:"compoundQuery",                               **"@type"**:"ns3:HpcCompoundMetadataQuery"                            },                            {                                 **"@minOccurs"**:"0",                               **"@name"**:"detailedResponse",                               **"@type"**:"xs:boolean"                            },                            {                                 **"@minOccurs"**:"0",                               **"@name"**:"page",                               **"@type"**:"xs:int"                            }                         ]                      }                   }                },                {                     **"@name"**:"HpcDataManagementModelDTO",                   **"xs.complexType"**:{                        **"xs.sequence"**:{                           **"xs.element"**:[                              {                                 **"@maxOccurs"**:"unbounded",                               **"@minOccurs"**:"0",                               **"@name"**:"collectionMetadataValidationRules",                               **"@nillable"**:"true",                               **"@type"**:"ns3:HpcMetadataValidationRule"                            },                            {                                 **"@maxOccurs"**:"unbounded",                               **"@minOccurs"**:"0",                               **"@name"**:"dataObjectMetadataValidationRules",                               **"@nillable"**:"true",                               **"@type"**:"ns3:HpcMetadataValidationRule"                            },                            {                                 **"@name"**:"dataHierarchy",                               **"@type"**:"ns1:HpcDataHierarchy"                            }                         ]                      }                   }                },                {                     **"@name"**:"HpcDataObjectDownloadRequestDTO",                   **"xs.complexType"**:{                        **"xs.sequence"**:{                           **"xs.element"**:{                              **"@name"**:"destination",                            **"@type"**:"ns2:HpcFileLocation"                         }                      }                   }                },                {                     **"@name"**:"HpcDataObjectDownloadResponseDTO",                   **"xs.complexType"**:{                        **"xs.sequence"**:{                           **"xs.element"**:[                              {                                 **"@name"**:"requestId",                               **"@type"**:"xs:string"                            },                            {                                 **"@name"**:"destinationLocation",                               **"@type"**:"ns2:HpcFileLocation"                            },                            {                                 **"@name"**:"destinationFile",                               **"@type"**:"xs:string"                            }                         ]                      }                   }                },                {                     **"@name"**:"HpcDataObjectListDTO",                   **"xs.complexType"**:{                        **"xs.sequence"**:{                           **"xs.element"**:[                              {                                 **"@maxOccurs"**:"unbounded",                               **"@minOccurs"**:"0",                               **"@name"**:"dataObjects",                               **"@nillable"**:"true",                               **"@type"**:"tns:HpcDataObjectDTO"                            },                            {                                 **"@maxOccurs"**:"unbounded",                               **"@minOccurs"**:"0",                               **"@name"**:"dataObjectPaths",                               **"@nillable"**:"true",                               **"@type"**:"xs:string"                            },                            {                                 **"@minOccurs"**:"0",                               **"@name"**:"page",                               **"@type"**:"xs:int"                            },                            {                                 **"@minOccurs"**:"0",                               **"@name"**:"limit",                               **"@type"**:"xs:int"                            }                         ]                      }                   }                },                {                     **"@name"**:"HpcDataObjectRegistrationDTO",                   **"xs.complexType"**:{                        **"xs.sequence"**:{                           **"xs.element"**:[                              {                                 **"@name"**:"source",                               **"@type"**:"ns2:HpcFileLocation"                            },                            {                                 **"@name"**:"callerObjectId",                               **"@type"**:"xs:string"                            },                            {                                 **"@maxOccurs"**:"unbounded",                               **"@name"**:"metadataEntries",                               **"@type"**:"ns3:HpcMetadataEntry"                            }                         ]                      }                   }                },                {                     **"@name"**:"HpcEntityPermissionRequestDTO",                   **"xs.complexType"**:{                        **"xs.sequence"**:{                           **"xs.element"**:[                              {                                 **"@name"**:"path",                               **"@type"**:"xs:string"                            },                            {                                 **"@maxOccurs"**:"unbounded",                               **"@minOccurs"**:"0",                               **"@name"**:"userPermissions",                               **"@nillable"**:"true",                               **"@type"**:"ns1:HpcUserPermission"                            },                            {                                 **"@maxOccurs"**:"unbounded",                               **"@minOccurs"**:"0",                               **"@name"**:"groupPermissions",                               **"@nillable"**:"true",                               **"@type"**:"ns1:HpcGroupPermission"                            }                         ]                      }                   }                },                {                     **"@name"**:"HpcEntityPermissionResponseListDTO",                   **"xs.complexType"**:{                        **"xs.sequence"**:{                           **"xs.element"**:{                              **"@maxOccurs"**:"unbounded",                            **"@minOccurs"**:"0",                            **"@name"**:"entityPermissionResponses",                            **"@nillable"**:"true",                            **"@type"**:"tns:HpcEntityPermissionResponseDTO"                         }                      }                   }                },                {                     **"@name"**:"HpcMetadataAttributesListDTO",                   **"xs.complexType"**:{                        **"xs.sequence"**:{                           **"xs.element"**:[                              {                                 **"@maxOccurs"**:"unbounded",                               **"@minOccurs"**:"0",                               **"@name"**:"collectionMetadataAttributes",                               **"@nillable"**:"true",                               **"@type"**:"ns3:HpcMetadataLevelAttributes"                            },                            {                                 **"@maxOccurs"**:"unbounded",                               **"@minOccurs"**:"0",                               **"@name"**:"dataObjectMetadataAttributes",                               **"@nillable"**:"true",                               **"@type"**:"ns3:HpcMetadataLevelAttributes"                            }                         ]                      }                   }                },                {                     **"@name"**:"HpcNamedCompoundMetadataQueryListDTO",                   **"xs.complexType"**:{                        **"xs.sequence"**:{                           **"xs.element"**:{                              **"@maxOccurs"**:"unbounded",                            **"@minOccurs"**:"0",                            **"@name"**:"queries",                            **"@nillable"**:"true",                            **"@type"**:"ns3:HpcNamedCompoundMetadataQuery"                         }                      }                   }                },                {                     **"@name"**:"HpcEntityPermissionRequestDTOs",                   **"xs.complexType"**:{                        **"xs.sequence"**:{                           **"xs.element"**:{                              **"@maxOccurs"**:"unbounded",                            **"@minOccurs"**:"0",                            **"@ref"**:"tns:HpcEntityPermissionRequestDTO"                         }                      }                   }                }             ],             **"xs.complexType"**:[                  {                     **"@name"**:"HpcCollectionDTO",                   **"xs.sequence"**:{                        **"xs.element"**:[                           {                              **"@name"**:"collection",                            **"@type"**:"ns1:HpcCollection"                         },                         {                              **"@name"**:"metadataEntries",                            **"@type"**:"ns3:HpcMetadataEntries"                         }                      ]                   }                },                {                     **"@name"**:"HpcDataObjectDTO",                   **"xs.sequence"**:{                        **"xs.element"**:[                           {                              **"@name"**:"dataObject",                            **"@type"**:"ns1:HpcDataObject"                         },                         {                              **"@name"**:"metadataEntries",                            **"@type"**:"ns3:HpcMetadataEntries"                         },                         {                              **"@minOccurs"**:"0",                            **"@name"**:"transferPercentCompletion",                            **"@type"**:"xs:string"                         }                      ]                   }                },                {                     **"@name"**:"HpcEntityPermissionResponseDTO",                   **"xs.sequence"**:{                        **"xs.element"**:[                           {                              **"@name"**:"path",                            **"@type"**:"xs:string"                         },                         {                              **"@maxOccurs"**:"unbounded",                            **"@minOccurs"**:"0",                            **"@name"**:"userPermissionResponses",                            **"@nillable"**:"true",                            **"@type"**:"tns:HpcUserPermissionResponseDTO"                         },                         {                              **"@maxOccurs"**:"unbounded",                            **"@minOccurs"**:"0",                            **"@name"**:"groupPermissionResponses",                            **"@nillable"**:"true",                            **"@type"**:"tns:HpcGroupPermissionResponseDTO"                         }                      ]                   }                },                {                     **"@name"**:"HpcUserPermissionResponseDTO",                   **"xs.sequence"**:{                        **"xs.element"**:[                           {                              **"@name"**:"userId",                            **"@type"**:"xs:string"                         },                         {                              **"@name"**:"result",                            **"@type"**:"xs:boolean"                         },                         {                              **"@name"**:"message",                            **"@type"**:"xs:string"                         }                      ]                   }                },                {                     **"@name"**:"HpcGroupPermissionResponseDTO",                   **"xs.sequence"**:{                        **"xs.element"**:[                           {                              **"@name"**:"groupId",                            **"@type"**:"xs:string"                         },                         {                              **"@name"**:"result",                            **"@type"**:"xs:boolean"                         },                         {                              **"@name"**:"message",                            **"@type"**:"xs:string"                         }                      ]                   }                }             ]          },          {               **"@attributeFormDefault"**:"unqualified",             **"@elementFormDefault"**:"unqualified",             **"@targetNamespace"**:"http://hpc.nci.nih.gov/domain/datamanagement",             **"xs.complexType"**:[                  {                     **"@name"**:"HpcUserPermission",                   **"xs.complexContent"**:{                        **"xs.extension"**:{                           **"@base"**:"tns:HpcEntityPermission",                         **"xs.sequence"**:{                              **"xs.element"**:{                                 **"@name"**:"userId",                               **"@type"**:"xs:string"                            }                         }                      }                   }                },                {                     **"@name"**:"HpcEntityPermission",                   **"xs.sequence"**:{                        **"xs.element"**:{                           **"@name"**:"permission",                         **"@type"**:"xs:string"                      }                   }                },                {                     **"@name"**:"HpcGroupPermission",                   **"xs.complexContent"**:{                        **"xs.extension"**:{                           **"@base"**:"tns:HpcEntityPermission",                         **"xs.sequence"**:{                              **"xs.element"**:{                                 **"@name"**:"groupId",                               **"@type"**:"xs:string"                            }                         }                      }                   }                },                {                     **"@name"**:"HpcCollection",                   **"xs.sequence"**:{                        **"xs.element"**:[                           {                              **"@name"**:"collectionId",                            **"@type"**:"xs:int"                         },                         {                              **"@name"**:"collectionName",                            **"@type"**:"xs:string"                         },                         {                              **"@name"**:"absolutePath",                            **"@type"**:"xs:string"                         },                         {                              **"@name"**:"collectionParentName",                            **"@type"**:"xs:string"                         },                         {                              **"@name"**:"collectionOwnerName",                            **"@type"**:"xs:string"                         },                         {                              **"@name"**:"collectionOwnerZone",                            **"@type"**:"xs:string"                         },                         {                              **"@name"**:"collectionMapId",                            **"@type"**:"xs:string"                         },                         {                              **"@name"**:"collectionInheritance",                            **"@type"**:"xs:string"                         },                         {                              **"@name"**:"comments",                            **"@type"**:"xs:string"                         },                         {                              **"@name"**:"info1",                            **"@type"**:"xs:string"                         },                         {                              **"@name"**:"info2",                            **"@type"**:"xs:string"                         },                         {                              **"@name"**:"createdAt",                            **"@type"**:"xs:dateTime"                         },                         {                              **"@name"**:"modifiedAt",                            **"@type"**:"xs:dateTime"                         },                         {                              **"@name"**:"specColType",                            **"@type"**:"xs:string"                         }                      ]                   }                },                {                     **"@name"**:"HpcDataObject",                   **"xs.sequence"**:{                        **"xs.element"**:[                           {                              **"@name"**:"id",                            **"@type"**:"xs:int"                         },                         {                              **"@name"**:"collectionId",                            **"@type"**:"xs:int"                         },                         {                              **"@name"**:"dataName",                            **"@type"**:"xs:string"                         },                         {                              **"@name"**:"collectionName",                            **"@type"**:"xs:string"                         },                         {                              **"@name"**:"absolutePath",                            **"@type"**:"xs:string"                         },                         {                              **"@name"**:"dataReplicationNumber",                            **"@type"**:"xs:int"                         },                         {                              **"@name"**:"dataVersion",                            **"@type"**:"xs:int"                         },                         {                              **"@name"**:"dataSize",                            **"@type"**:"xs:long"                         },                         {                              **"@name"**:"dataTypeName",                            **"@type"**:"xs:string"                         },                         {                              **"@name"**:"resourceGroupName",                            **"@type"**:"xs:string"                         },                         {                              **"@name"**:"resourceName",                            **"@type"**:"xs:string"                         },                         {                              **"@name"**:"dataPath",                            **"@type"**:"xs:string"                         },                         {                              **"@name"**:"dataOwnerName",                            **"@type"**:"xs:string"                         },                         {                              **"@name"**:"dataOwnerZone",                            **"@type"**:"xs:string"                         },                         {                              **"@name"**:"replicationStatus",                            **"@type"**:"xs:string"                         },                         {                              **"@name"**:"dataStatus",                            **"@type"**:"xs:string"                         },                         {                              **"@name"**:"checksum",                            **"@type"**:"xs:string"                         },                         {                              **"@name"**:"expiry",                            **"@type"**:"xs:string"                         },                         {                              **"@name"**:"dataMapId",                            **"@type"**:"xs:int"                         },                         {                              **"@name"**:"comments",                            **"@type"**:"xs:string"                         },                         {                              **"@name"**:"createdAt",                            **"@type"**:"xs:dateTime"                         },                         {                              **"@name"**:"updatedAt",                            **"@type"**:"xs:dateTime"                         },                         {                              **"@name"**:"specColType",                            **"@type"**:"xs:string"                         }                      ]                   }                },                {                     **"@name"**:"HpcDataHierarchy",                   **"xs.sequence"**:{                        **"xs.element"**:[                           {                              **"@name"**:"collectionType",                            **"@type"**:"xs:string"                         },                         {                              **"@name"**:"isDataObjectContainer",                            **"@type"**:"xs:boolean"                         },                         {                              **"@maxOccurs"**:"unbounded",                            **"@minOccurs"**:"0",                            **"@name"**:"subCollectionsHierarchies",                            **"@nillable"**:"true",                            **"@type"**:"tns:HpcDataHierarchy"                         }                      ]                   }                }             ]          },          {               **"@attributeFormDefault"**:"unqualified",             **"@elementFormDefault"**:"unqualified",             **"@targetNamespace"**:"http://hpc.nci.nih.gov/domain/notification",             **"xs.complexType"**:{                  **"@name"**:"HpcNotificationSubscription",                **"xs.sequence"**:{                     **"xs.element"**:[                        {                           **"@name"**:"eventType",                         **"@type"**:"xs:string"                      },                      {                           **"@maxOccurs"**:"unbounded",                         **"@minOccurs"**:"0",                         **"@name"**:"notificationDeliveryMethods",                         **"@nillable"**:"true",                         **"@type"**:"xs:string"                      }                   ]                }             },             **"xs.simpleType"**:[                  {                     **"@name"**:"HpcEventType",                   **"xs.restriction"**:{                        **"@base"**:"xs:string",                      **"xs.enumeration"**:[                           {                              **"@value"**:"DATA\_TRANSFER\_UPLOAD\_IN\_TEMPORARY\_ARCHIVE"                         },                         {                              **"@value"**:"DATA\_TRANSFER\_UPLOAD\_ARCHIVED"                         },                         {                              **"@value"**:"DATA\_TRANSFER\_UPLOAD\_FAILED"                         },                         {                              **"@value"**:"DATA\_TRANSFER\_DOWNLOAD\_COMPLETED"                         },                         {                              **"@value"**:"DATA\_TRANSFER\_DOWNLOAD\_FAILED"                         },                         {                              **"@value"**:"USAGE\_SUMMARY\_REPORT"                         },                         {                              **"@value"**:"USAGE\_SUMMARY\_BY\_WEEKLY\_REPORT"                         },                         {                              **"@value"**:"USAGE\_SUMMARY\_BY\_DOC\_REPORT"                         },                         {                              **"@value"**:"USAGE\_SUMMARY\_BY\_DOC\_BY\_WEEKLY\_REPORT"                         },                         {                              **"@value"**:"USAGE\_SUMMARY\_BY\_USER\_REPORT"                         },                         {                              **"@value"**:"USAGE\_SUMMARY\_BY\_USER\_BY\_WEEKLY\_REPORT"                         }                      ]                   }                },                {                     **"@name"**:"HpcNotificationDeliveryMethod",                   **"xs.restriction"**:{                        **"@base"**:"xs:string",                      **"xs.enumeration"**:[                           {                              **"@value"**:"EMAIL"                         },                         {                              **"@value"**:"TEXT"                         }                      ]                   }                }             ]          }       ]    },    **"resources"**:{         **"@base"**:"https://fr-s-hpcdm-gp-d.ncifcrf.gov:7738/hpc-server",       **"resource"**:[            {               **"@path"**:"/",             **"resource"**:[                  {                     **"@path"**:"authenticate",                   **"method"**:{                        **"@name"**:"GET",                      **"response"**:{                           **"representation"**:[                              {                                 **"@element"**:"Response",                               **"@mediaType"**:"application/json"                            },                            {                                 **"@mediaType"**:"application/xml"                            }                         ]                      }                   }                },                {                     **"@path"**:"group",                   **"method"**:{                        **"@name"**:"POST",                      **"request"**:{                           **"representation"**:[                              {                                 **"@element"**:"HpcGroupRequestDTO",                               **"@mediaType"**:"application/json"                            },                            {                                 **"@mediaType"**:"application/xml"                            }                         ]                      },                      **"response"**:{                           **"representation"**:[                              {                                 **"@element"**:"Response",                               **"@mediaType"**:"application/json"                            },                            {                                 **"@mediaType"**:"application/xml"                            }                         ]                      }                   }                },                {                     **"@path"**:"systemAccount",                   **"method"**:{                        **"@name"**:"PUT",                      **"request"**:{                           **"representation"**:[                              {                                 **"@element"**:"HpcSystemAccountDTO",                               **"@mediaType"**:"application/json"                            },                            {                                 **"@mediaType"**:"application/xml"                            }                         ]                      },                      **"response"**:{                           **"representation"**:{                              **"@mediaType"**:"\*/\*"                         }                      }                   }                },                {                     **"@path"**:"user",                   **"method"**:{                        **"@name"**:"PUT",                      **"request"**:{                           **"representation"**:[                              {                                 **"@element"**:"HpcUserDTO",                               **"@mediaType"**:"application/json"                            },                            {                                 **"@mediaType"**:"application/xml"                            }                         ]                      },                      **"response"**:{                           **"representation"**:{                              **"@mediaType"**:"\*/\*"                         }                      }                   }                },                {                     **"@path"**:"user/{nciUserId}",                   **"param"**:{                        **"@name"**:"nciUserId",                      **"@style"**:"template",                      **"@type"**:"string"                   },                   **"method"**:[                        {                           **"@name"**:"GET",                         **"request"**:"",                         **"response"**:{                              **"representation"**:[                                 {                                    **"@element"**:"Response",                                  **"@mediaType"**:"application/json"                               },                               {                                    **"@mediaType"**:"application/xml"                               }                            ]                         }                      },                      {                           **"@name"**:"POST",                         **"request"**:{                              **"representation"**:[                                 {                                    **"@element"**:"HpcUpdateUserRequestDTO",                                  **"@mediaType"**:"application/json"                               },                               {                                    **"@mediaType"**:"application/xml"                               }                            ]                         },                         **"response"**:{                              **"representation"**:{                                 **"@mediaType"**:"\*/\*"                            }                         }                      }                   ]                }             ]          },          {               **"@path"**:"/",             **"resource"**:[                  {                     **"@path"**:"acl",                   **"method"**:{                        **"@name"**:"POST",                      **"request"**:{                           **"representation"**:[                              {                                 **"@element"**:"HpcEntityPermissionRequestDTO",                               **"@mediaType"**:"application/json"                            },                            {                                 **"@mediaType"**:"application/xml"                            }                         ]                      },                      **"response"**:{                           **"representation"**:[                              {                                 **"@element"**:"Response",                               **"@mediaType"**:"application/json"                            },                            {                                 **"@mediaType"**:"application/xml"                            }                         ]                      }                   }                },                {                     **"@path"**:"collection/{path:.\*}",                   **"param"**:{                        **"@name"**:"path",                      **"@style"**:"template",                      **"@type"**:"string"                   },                   **"method"**:[                        {                           **"@name"**:"GET",                         **"request"**:"",                         **"response"**:{                              **"representation"**:[                                 {                                    **"@element"**:"Response",                                  **"@mediaType"**:"application/json"                               },                               {                                    **"@mediaType"**:"application/xml"                               }                            ]                         }                      },                      {                           **"@name"**:"PUT",                         **"request"**:{                              **"representation"**:[                                 {                                    **"@element"**:"HpcMetadataEntry",                                  **"@mediaType"**:"application/json"                               },                               {                                    **"@mediaType"**:"application/xml"                               }                            ]                         },                         **"response"**:{                              **"representation"**:[                                 {                                    **"@element"**:"Response",                                  **"@mediaType"**:"application/json"                               },                               {                                    **"@mediaType"**:"application/xml"                               }                            ]                         }                      }                   ]                },                {                     **"@path"**:"dataObject/{path:.\*}",                   **"param"**:{                        **"@name"**:"path",                      **"@style"**:"template",                      **"@type"**:"string"                   },                   **"method"**:[                        {                           **"@name"**:"GET",                         **"request"**:"",                         **"response"**:{                              **"representation"**:[                                 {                                    **"@element"**:"Response",                                  **"@mediaType"**:"application/json"                               },                               {                                    **"@mediaType"**:"application/xml"                               }                            ]                         }                      },                      {                           **"@name"**:"PUT",                         **"request"**:{                              **"representation"**:{                                 **"@mediaType"**:"multipart/form-data"                            }                         },                         **"response"**:{                              **"representation"**:[                                 {                                    **"@element"**:"Response",                                  **"@mediaType"**:"application/json"                               },                               {                                    **"@mediaType"**:"application/xml"                               }                            ]                         }                      }                   ]                },                {                     **"@path"**:"dataObject/{path:.\*}/download",                   **"param"**:{                        **"@name"**:"path",                      **"@style"**:"template",                      **"@type"**:"string"                   },                   **"method"**:{                        **"@name"**:"POST",                      **"request"**:{                           **"representation"**:[                              {                                 **"@element"**:"HpcDataObjectDownloadRequestDTO",                               **"@mediaType"**:"application/json"                            },                            {                                 **"@mediaType"**:"application/xml"                            }                         ]                      },                      **"response"**:{                           **"representation"**:[                              {                                 **"@element"**:"Response",                               **"@mediaType"**:"application/json"                            },                            {                                 **"@mediaType"**:"application/xml"                            },                            {                                 **"@mediaType"**:"application/octet-stream"                            }                         ]                      }                   }                },                {                     **"@path"**:"model/{doc}",                   **"param"**:{                        **"@name"**:"doc",                      **"@style"**:"template",                      **"@type"**:"string"                   },                   **"method"**:{                        **"@name"**:"GET",                      **"request"**:"",                      **"response"**:{                           **"representation"**:[                              {                                 **"@element"**:"Response",                               **"@mediaType"**:"application/json"                            },                            {                                 **"@mediaType"**:"application/xml"                            }                         ]                      }                   }                }             ]          },          {               **"@path"**:"/",             **"resource"**:[                  {                     **"@path"**:"collection/query",                   **"method"**:{                        **"@name"**:"POST",                      **"request"**:{                           **"representation"**:[                              {                                 **"@element"**:"HpcCompoundMetadataQueryDTO",                               **"@mediaType"**:"application/json"                            },                            {                                 **"@mediaType"**:"application/xml"                            }                         ]                      },                      **"response"**:{                           **"representation"**:[                              {                                 **"@element"**:"Response",                               **"@mediaType"**:"application/json"                            },                            {                                 **"@mediaType"**:"application/xml"                            }                         ]                      }                   }                },                {                     **"@path"**:"collection/query/{queryName}",                   **"param"**:{                        **"@name"**:"queryName",                      **"@style"**:"template",                      **"@type"**:"string"                   },                   **"method"**:{                        **"@name"**:"GET",                      **"request"**:{                           **"param"**:[                              {                                 **"@name"**:"detailedResponse",                               **"@style"**:"query",                               **"@type"**:"boolean"                            },                            {                                 **"@name"**:"page",                               **"@style"**:"query",                               **"@type"**:"int"                            }                         ]                      },                      **"response"**:{                           **"representation"**:[                              {                                 **"@element"**:"Response",                               **"@mediaType"**:"application/json"                            },                            {                                 **"@mediaType"**:"application/xml"                            }                         ]                      }                   }                },                {                     **"@path"**:"dataObject/query",                   **"method"**:{                        **"@name"**:"POST",                      **"request"**:{                           **"representation"**:[                              {                                 **"@element"**:"HpcCompoundMetadataQueryDTO",                               **"@mediaType"**:"application/json"                            },                            {                                 **"@mediaType"**:"application/xml"                            }                         ]                      },                      **"response"**:{                           **"representation"**:[                              {                                 **"@element"**:"Response",                               **"@mediaType"**:"application/json"                            },                            {                                 **"@mediaType"**:"application/xml"                            }                         ]                      }                   }                },                {                     **"@path"**:"dataObject/query/{queryName}",                   **"param"**:{                        **"@name"**:"queryName",                      **"@style"**:"template",                      **"@type"**:"string"                   },                   **"method"**:{                        **"@name"**:"GET",                      **"request"**:{                           **"param"**:[                              {                                 **"@name"**:"detailedResponse",                               **"@style"**:"query",                               **"@type"**:"boolean"                            },                            {                                 **"@name"**:"page",                               **"@style"**:"query",                               **"@type"**:"int"                            }                         ]                      },                      **"response"**:{                           **"representation"**:[                              {                                 **"@element"**:"Response",                               **"@mediaType"**:"application/json"                            },                            {                                 **"@mediaType"**:"application/xml"                            }                         ]                      }                   }                },                {                     **"@path"**:"metadataAttributes",                   **"method"**:{                        **"@name"**:"GET",                      **"request"**:{                           **"param"**:[                              {                                 **"@name"**:"level",                               **"@style"**:"query",                               **"@type"**:"int"                            },                            {                                 **"@name"**:"levelOperator",                               **"@style"**:"query",                               **"@type"**:"string"                            }                         ]                      },                      **"response"**:{                           **"representation"**:[                              {                                 **"@element"**:"Response",                               **"@mediaType"**:"application/json"                            },                            {                                 **"@mediaType"**:"application/xml"                            }                         ]                      }                   }                },                {                     **"@path"**:"query",                   **"method"**:{                        **"@name"**:"GET",                      **"response"**:{                           **"representation"**:[                              {                                 **"@element"**:"Response",                               **"@mediaType"**:"application/json"                            },                            {                                 **"@mediaType"**:"application/xml"                            }                         ]                      }                   }                },                {                     **"@path"**:"query/{queryName}",                   **"param"**:{                        **"@name"**:"queryName",                      **"@style"**:"template",                      **"@type"**:"string"                   },                   **"method"**:[                        {                           **"@name"**:"DELETE",                         **"request"**:"",                         **"response"**:{                              **"representation"**:{                                 **"@mediaType"**:"\*/\*"                            }                         }                      },                      {                           **"@name"**:"POST",                         **"request"**:{                              **"representation"**:[                                 {                                    **"@element"**:"HpcCompoundMetadataQueryDTO",                                  **"@mediaType"**:"application/json"                               },                               {                                    **"@mediaType"**:"application/xml"                               }                            ]                         },                         **"response"**:{                              **"representation"**:{                                 **"@mediaType"**:"\*/\*"                            }                         }                      }                   ]                },                {                     **"@path"**:"refreshMetadataViews",                   **"method"**:{                        **"@name"**:"POST",                      **"response"**:{                           **"representation"**:{                              **"@mediaType"**:"\*/\*"                         }                      }                   }                }             ]          },          {               **"@path"**:"/",             **"resource"**:{                  **"@path"**:"notification/{nciUserId}",                **"param"**:{                     **"@name"**:"nciUserId",                   **"@style"**:"template",                   **"@type"**:"string"                },                **"method"**:[                     {                        **"@name"**:"GET",                      **"request"**:"",                      **"response"**:{                           **"representation"**:[                              {                                 **"@element"**:"Response",                               **"@mediaType"**:"application/json"                            },                            {                                 **"@mediaType"**:"application/xml"                            }                         ]                      }                   },                   {                        **"@name"**:"POST",                      **"request"**:{                           **"representation"**:[                              {                                 **"@element"**:"HpcNotificationSubscriptionsRequestDTO",                               **"@mediaType"**:"application/json"                            },                            {                                 **"@mediaType"**:"application/xml"                            }                         ]                      },                      **"response"**:{                           **"representation"**:[                              {                                 **"@element"**:"Response",                               **"@mediaType"**:"application/json"                            },                            {                                 **"@mediaType"**:"application/xml"                            }                         ]                      }                   }                ]             }          },          {               **"@path"**:"/",             **"resource"**:{                  **"@path"**:"report",                **"method"**:{                     **"@name"**:"POST",                   **"request"**:{                        **"representation"**:[                           {                              **"@element"**:"HpcReportRequestDTO",                            **"@mediaType"**:"application/json"                         },                         {                              **"@mediaType"**:"application/xml"                         }                      ]                   },                   **"response"**:{                        **"representation"**:[                           {                              **"@element"**:"Response",                            **"@mediaType"**:"application/json"                         },                         {                              **"@mediaType"**:"application/xml"                         }                      ]                   }                }             }          }       ]    } } |