MoDaC REST API Specification

Version *1.0*

*01/28/2021*

Table of Contents

[1. Purpose 3](#_Toc62833160)

[2. Introduction 3](#_Toc62833161)

[2.1 What is MoDaC? 3](#_Toc62833162)

[2.2 MoDaC URL 3](#_Toc62833163)

[3. MoDaC Server API Specification 3](#_Toc62833164)

[3.1 Register Collection 3](#_Toc62833165)

[3.2 Get Collection 9](#_Toc62833166)

[3.3 Find Collection by compound metadata query 15](#_Toc62833167)

[3.4 Find data by compound metadata query 28](#_Toc62833168)

[3.5 Get Data File (V2) 40](#_Toc62833169)

[3.6 Download Data File (v2) 46](#_Toc62833170)

[3.7 Download Collection (V2) 49](#_Toc62833171)

[3.8 Download Collection List 51](#_Toc62833172)

[3.9 Download Data Object List (V2) 54](#_Toc62833173)

[3.10 Register Data File (V2) 56](#_Toc62833174)

[3.11 Generate Download Request URL 67](#_Toc62833175)

**Version History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Document Version Number** | **Software Release Number** | **Revision**  **Date** | **Revision Description** |
| 1.0 | 1.3 | 1/28/2021 | Initial version |

# Purpose

This document describes the REST APIs available to users of the Predictive Oncology Model and Data Clearinghouse (MoDaC). The specification provides the format of the REST APIs for programmatically accessing the capabilities of the platform including searching, downloading, and uploading datasets and software models to MoDaC.

# Introduction

# What is MoDaC?

The Predictive Oncology Model and Data Clearinghouse (MoDaC) is a data-sharing repository developed to transition assets to the broader research community. These assets include datasets and software models from computational capabilities developed within NCI and in collaborative programs, including the [Joint Design of Advanced Computing Solutions for Cancer (JDACS4C) Program](https://datascience.cancer.gov/collaborations/joint-design-advanced-computing) and the [Accelerating Therapeutics for Opportunities in Medicine (ATOM) Consortium](https://datascience.cancer.gov/collaborations/atom).

## MoDaC URL

The MoDaC Server API URL: [https://modac.cancer.gov](https://modac.cancer.gov/)

# MoDaC Server API Specification

## Register Collection

Collections support logical hierarchical structure where a collection can have a parent and many children. 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. The service expects the containing collection for the registered collection to exist at time of registration.

|  |  |
| --- | --- |
| Title | Register Collection |
| Description | This registers a collection. Presently, 3 types of collections are supported in hierarchical order– Program, Study and Asset. The type of the collection is indicated in the collection\_type field of the JSON metadata file. |
| URL | /api/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": “MoDaC"  },  {  "attribute": "original\_date\_created",  "value": "1/28/2021"  }    ],  “createParentCollections” : true,    “parentCollectionsBulkMetadataEntires”: {  “defaultCollectionMetadataEntries”: [  {  "attribute": "pi\_name",  "value": "Udit’s PI LAB GLB"  }  ],  "pathsMetadataEntries": [  {  "path": "/NCI\_DOE\_Archive/JDACS4C/JDACS4C\_Pilot\_2/Sehgal\_Simulation\_Data",  "pathMetadataEntries": [  {  "attribute": "pi\_name",  "value": "Eran PI LAB GLB"  }  ]  }  ]  }  ]  }  }  **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>MoDaC</value>  </element>  <element>  <attribute>original\_date\_created</attribute>  <value>10/25/2006</value>  </element>  </metadataEntries> |
| Success Response | HTTP/1.1 201 Created  The following is the successful response on updating an existing Collection  HTTP/1.1 200 OK |
| Error Response | **Invalid Request Input:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  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  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  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  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  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  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  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  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  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": "…"  } |

## Get Collection

|  |  |
| --- | --- |
| Title | Get Collection |
| Description | Get information about a specific collection including all of its hierarachical metadata |
| URL | /api/collection/{path}?list=[true|false] |
| Method | GET |
| Acceptable request representation | application/json  application/xml |
| Available response representation |  |
| URL Params | path – The collection path |
| Query params | list – If set to true, the API will list the child collections and data objects of this collection. This is an optional parameter and default is false  includeAcl – If set to true, the API will provide the permission of the requesting user to this collection (READ, WRITE or OWN). This is an optional parameter and the default is false |
| Data Params |  |
| Success Response | **Example 1 - without query params:**  HTTP/1.1 200 OK  {  "collections": [{  "collection": {  "collectionId": 6282617,  "collectionName": "/NCI\_DOE\_Archive/ATOM",  "absolutePath": "/NCI\_DOE\_Archive/ATOM",  "collectionParentName": "/NCI\_DOE\_Archive",  "collectionOwnerName": "ncidoesvcp2",  "collectionOwnerZone": "ncifprodZone",  "collectionMapId": "0",  "collectionInheritance": "1",  "comments": "",  "info1": "",  "info2": "",  "createdAt": "2020-09-18T20:56:10.000+00:00",  "specColType": "NORMAL",  "subCollections": [],  "dataObjects": []  },  "metadataEntries": {  "selfMetadataEntries": [{  "attribute": "access\_group",  "value": "public",  "unit": null,  "level": null,  "levelLabel": null,  "collectionId": null  }, {  "attribute": "program\_name",  "value": "Accelerating Therapeutics for Opportunities in Medicine (ATOM)",  "unit": null,  "level": null,  "levelLabel": null,  "collectionId": null  }, {  "attribute": "uuid",  "value": "2eba0f8c-0a21-469f-b5e4-13bfcb43dc21",  "unit": null,  "level": null,  "levelLabel": null,  "collectionId": null  }, {  "attribute": "registered\_by\_name",  "value": "NCI\_DOE NCI\_DOE",  "unit": null,  "level": null,  "levelLabel": null,  "collectionId": null  }, {  "attribute": "registered\_by",  "value": "ncidoesvcp2",  "unit": null,  "level": null,  "levelLabel": null,  "collectionId": null  }, {  "attribute": "collection\_type",  "value": "Program",  "unit": null,  "level": null,  "levelLabel": null,  "collectionId": null  }, {  "attribute": "program\_identifier",  "value": "ATOM",  "unit": null,  "level": null,  "levelLabel": null,  "collectionId": null  }, {  "attribute": "program\_description",  "value": "The Accelerating Therapeutics for Opportunities in Medicine (ATOM) Consortium is a public-private partnership whose mission is to transform drug discovery by accelerating the development of more effective therapies for patients. It is comprised of a partnership between the Frederick National Laboratory for Cancer Research, GSK, Lawrence Livermore National Laboratory, and the University of California, San Francisco.",  "unit": null,  "level": null,  "levelLabel": null,  "collectionId": null  }, {  "attribute": "configuration\_id",  "value": "4ac6a3d2-48bf-4a2e-8374-fe5449561d9b",  "unit": null,  "level": null,  "levelLabel": null,  "collectionId": null  }, {  "attribute": "metadata\_updated",  "value": "11-10-2020 16:46:44",  "unit": null,  "level": null,  "levelLabel": null,  "collectionId": null  }],  "parentMetadataEntries": []  },  "permission": null  }],  "collectionPaths": [],  "page": null,  "limit": null,  "totalCount": null  }  **Example 2 - with query params ( list and includeAcl) set to true):**  {  "collections": [{  "collection": {  "collectionId": 6282617,  "collectionName": "/NCI\_DOE\_Archive/ATOM",  "absolutePath": "/NCI\_DOE\_Archive/ATOM",  "collectionParentName": "/NCI\_DOE\_Archive",  "collectionOwnerName": "ncidoesvcp2",  "collectionOwnerZone": "ncifprodZone",  "collectionMapId": "0",  "collectionInheritance": "1",  "comments": "",  "info1": "",  "info2": "",  "createdAt": "2020-09-18T20:56:10.000+00:00",  "specColType": "NORMAL",  "subCollections": [{  "id": 6611412,  "path": "/NCI\_DOE\_Archive/ATOM/Neurocrine\_H1\_Demonstration",  "dataSize": 0,  "createdAt": null  }],  "dataObjects": []  },  "metadataEntries": {  "selfMetadataEntries": [{  "attribute": "access\_group",  "value": "public",  "unit": null,  "level": null,  "levelLabel": null,  "collectionId": null  }, {  "attribute": "program\_name",  "value": "Accelerating Therapeutics for Opportunities in Medicine (ATOM)",  "unit": null,  "level": null,  "levelLabel": null,  "collectionId": null  }, {  "attribute": "uuid",  "value": "2eba0f8c-0a21-469f-b5e4-13bfcb43dc21",  "unit": null,  "level": null,  "levelLabel": null,  "collectionId": null  }, {  "attribute": "registered\_by\_name",  "value": "NCI\_DOE NCI\_DOE",  "unit": null,  "level": null,  "levelLabel": null,  "collectionId": null  }, {  "attribute": "registered\_by",  "value": "ncidoesvcp2",  "unit": null,  "level": null,  "levelLabel": null,  "collectionId": null  }, {  "attribute": "collection\_type",  "value": "Program",  "unit": null,  "level": null,  "levelLabel": null,  "collectionId": null  }, {  "attribute": "program\_identifier",  "value": "ATOM",  "unit": null,  "level": null,  "levelLabel": null,  "collectionId": null  }, {  "attribute": "program\_description",  "value": "The Accelerating Therapeutics for Opportunities in Medicine (ATOM) Consortium is a public-private partnership whose mission is to transform drug discovery by accelerating the development of more effective therapies for patients. It is comprised of a partnership between the Frederick National Laboratory for Cancer Research, GSK, Lawrence Livermore National Laboratory, and the University of California, San Francisco.",  "unit": null,  "level": null,  "levelLabel": null,  "collectionId": null  }, {  "attribute": "configuration\_id",  "value": "4ac6a3d2-48bf-4a2e-8374-fe5449561d9b",  "unit": null,  "level": null,  "levelLabel": null,  "collectionId": null  }, {  "attribute": "metadata\_updated",  "value": "11-10-2020 16:46:44",  "unit": null,  "level": null,  "levelLabel": null,  "collectionId": null  }],  "parentMetadataEntries": []  },  "permission": "OWN"  }],  "collectionPaths": [],  "page": null,  "limit": null,  "totalCount": null  } |
| 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/1.1 401 Bad Request  Content-Type: application/json  {  "code": "Error occurred",  "message": "Not Authorized"  }  **Collection doesn’t exist:**  HTTP/1.1 204 No Content |

## 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, TIMESTAMP\_LESS\_THAN, TIMESTAMP\_GREATER\_THAN, TIMESTAMP\_LESS\_OR\_EQUAL, TIMESTAMP\_GREATER\_OR\_EQUAL | | | 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, TIMESTAMP\_LESS\_THAN, TIMESTAMP\_GREATER\_THAN, TIMESTAMP\_LESS\_OR\_EQUAL, TIMESTAMP\_GREATER\_OR\_EQUAL | | format | MM-DD-YYYY HH24:MI:SS  MM/DD/YY et.c | | | 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. | | | | NOTE: ‘format’ attribute is only applicable for TIMESTMP operators. | | |   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.  Examples:   |  | | --- | | **Example 1:**  {  "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": false,     "page" : 1,     "totalCount" : true  }  **Example 2:**  {     "compoundQuery":    {        "operator": "OR",        "queries": [  {  "attribute":"data\_transfer\_completed",  "value": "08-09-2017 22:18:26",  "format":"MM-DD-YYYHH24:MI:SS",  "operator" : "TIMESTAMP\_LESS\_THAN"  }     ]  },     "detailedResponse": false,     "page" : 1,     "totalCount" : true  } |   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": true,  "page": 1,  "totalCount": true  } |
| Success Response | **Example 1: - detailedResponse: false**  HTTP/1.1 200 OK  {  "collectionPaths": ["NCI\_DOE\_Archive/ATOM/Neurocrine\_H1\_Demonstration/H1\_Selectivity\_Assay "],  "page": 1,  "limit": 100  }  **Example 1: - detailedResponse: true**  HTTP/1.1 200 OK  Content-Type: application/json  {  "collections": [{  "collection": {  "collectionId": 6611544,  "collectionName": "/NCI\_DOE\_Archive/ATOM/Neurocrine\_H1\_Demonstration/H1\_Selectivity\_Assay",  "absolutePath": "/NCI\_DOE\_Archive/ATOM/Neurocrine\_H1\_Demonstration/H1\_Selectivity\_Assay",  "collectionParentName": "/NCI\_DOE\_Archive/ATOM/Neurocrine\_H1\_Demonstration",  "collectionOwnerName": "ncidoesvcp2",  "collectionOwnerZone": "ncifprodZone",  "collectionMapId": "0",  "collectionInheritance": "1",  "comments": null,  "info1": null,  "info2": null,  "createdAt": "2020-11-20T17:06:25.000+00:00",  "specColType": null,  "subCollections": [],  "dataObjects": []  },  "metadataEntries": {  "selfMetadataEntries": [{  "attribute": "access\_group",  "value": "public",  "unit": null,  "level": 1,  "levelLabel": "Asset",  "collectionId": 6611544  }, {  "attribute": "configuration\_id",  "value": "4ac6a3d2-48bf-4a2e-8374-fe5449561d9b",  "unit": null,  "level": 1,  "levelLabel": "Asset",  "collectionId": 6611544  }, {  "attribute": "registered\_by",  "value": "ncidoesvcp2",  "unit": null,  "level": 1,  "levelLabel": "Asset",  "collectionId": 6611544  },  {  "attribute": "uuid",  "value": "e497fd73-7f50-465d-a6be-557df828eae1",  "unit": null,  "level": 1,  "levelLabel": "Asset",  "collectionId": 6611544  }, {  "attribute": "metadata\_updated",  "value": "01-29-2021 00:23:04",  "unit": null,  "level": 1,  "levelLabel": "Asset",  "collectionId": 6611544  }, {  "attribute": "asset\_identifier",  "value": "H1\_Selectivity\_Assay",  "unit": null,  "level": 1,  "levelLabel": "Asset",  "collectionId": 6611544  }, {  "attribute": "collection\_type",  "value": "Asset",  "unit": null,  "level": 1,  "levelLabel": "Asset",  "collectionId": 6611544  }, {  "attribute": "description",  "value": "Histamine-1 (H1), Muscarinic Receptors 1-5 (M1-M5) and hERG binding affinity along with ligand structural data.",  "unit": null,  "level": 1,  "levelLabel": "Asset",  "collectionId": 6611544  }, {  "attribute": "asset\_name",  "value": "H1 Selectivity Assay",  "unit": null,  "level": 1,  "levelLabel": "Asset",  "collectionId": 6611544  }, {  "attribute": "asset\_type",  "value": "Dataset",  "unit": null,  "level": 1,  "levelLabel": "Asset",  "collectionId": 6611544  }, {  "attribute": "registered\_by\_name",  "value": "NCI\_DOE NCI\_DOE",  "unit": null,  "level": 1,  "levelLabel": "Asset",  "collectionId": 6611544  }],  "parentMetadataEntries": [{  "attribute": "study\_name",  "value": "Neurocrine H1 Demonstration",  "unit": null,  "level": 2,  "levelLabel": "Study",  "collectionId": 6611412  }, {  "attribute": "study\_identifier",  "value": "Neurocrine\_H1\_Demonstration",  "unit": null,  "level": 2,  "levelLabel": "Study",  "collectionId": 6611412  }, {  "attribute": "registered\_by",  "value": "ncidoesvcp2",  "unit": null,  "level": 2,  "levelLabel": "Study",  "collectionId": 6611412  }, {  "attribute": "access\_group",  "value": "public",  "unit": null,  "level": 2,  "levelLabel": "Study",  "collectionId": 6611412  }, {  "attribute": "description",  "value": "Design and synthesize selective H1 antagonists",  "unit": null,  "level": 2,  "levelLabel": "Study",  "collectionId": 6611412  },  {  "attribute": "collection\_type",  "value": "Study",  "unit": null,  "level": 2,  "levelLabel": "Study",  "collectionId": 6611412  }, {  "attribute": "configuration\_id",  "value": "4ac6a3d2-48bf-4a2e-8374-fe5449561d9b",  "unit": null,  "level": 2,  "levelLabel": "Study",  "collectionId": 6611412  }, {  "attribute": "metadata\_updated",  "value": "11-20-2020 12:01:58",  "unit": null,  "level": 2,  "levelLabel": "Study",  "collectionId": 6611412  }, {  "attribute": "uuid",  "value": "1ca4b087-5c76-4bc3-9354-2f3122500e45",  "unit": null,  "level": 2,  "levelLabel": "Study",  "collectionId": 6611412  }, {  "attribute": "registered\_by\_name",  "value": "NCI\_DOE NCI\_DOE",  "unit": null,  "level": 2,  "levelLabel": "Study",  "collectionId": 6611412  }, {  "attribute": "collection\_type",  "value": "Program",  "unit": null,  "level": 3,  "levelLabel": "Program",  "collectionId": 6282617  }, {  "attribute": "uuid",  "value": "2eba0f8c-0a21-469f-b5e4-13bfcb43dc21",  "unit": null,  "level": 3,  "levelLabel": "Program",  "collectionId": 6282617  }, {  "attribute": "access\_group",  "value": "public",  "unit": null,  "level": 3,  "levelLabel": "Program",  "collectionId": 6282617  }, {  "attribute": "configuration\_id",  "value": "4ac6a3d2-48bf-4a2e-8374-fe5449561d9b",  "unit": null,  "level": 3,  "levelLabel": "Program",  "collectionId": 6282617  }, {  "attribute": "registered\_by",  "value": "ncidoesvcp2",  "unit": null,  "level": 3,  "levelLabel": "Program",  "collectionId": 6282617  }, {  "attribute": "metadata\_updated",  "value": "11-10-2020 16:46:44",  "unit": null,  "level": 3,  "levelLabel": "Program",  "collectionId": 6282617  }, {  "attribute": "program\_name",  "value": "Accelerating Therapeutics for Opportunities in Medicine (ATOM)",  "unit": null,  "level": 3,  "levelLabel": "Program",  "collectionId": 6282617  }, {  "attribute": "program\_identifier",  "value": "ATOM",  "unit": null,  "level": 3,  "levelLabel": "Program",  "collectionId": 6282617  }, {  "attribute": "program\_description",  "value": "The Accelerating Therapeutics for Opportunities in Medicine (ATOM) Consortium is a public-private partnership whose mission is to transform drug discovery by accelerating the development of more effective therapies for patients. It is comprised of a partnership between the Frederick National Laboratory for Cancer Research, GSK, Lawrence Livermore National Laboratory, and the University of California, San Francisco.",  "unit": null,  "level": 3,  "levelLabel": "Program",  "collectionId": 6282617  }, {  "attribute": "registered\_by\_name",  "value": "NCI\_DOE NCI\_DOE",  "unit": null,  "level": 3,  "levelLabel": "Program",  "collectionId": 6282617  }]  },  "permission": null  }],  "collectionPaths": [],  "page": 1,  "limit": 100,  "totalCount": 1  } |
| Error Response | **Data Management Error:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  JSON:  {  "errorType": " DATABASE\_ERROR",  "message": " Failed to get Collections: ",  "stackTrace": "…"  }  **Invalid query attributes:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  {"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 |

## 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 | /api/dataObject/query |
| Method | POST |
| Acceptable request representation | application/json  application/xml |
| Available response representation | application/json  application/xml |
| URL Params | {urlPath} – Logical path of the collection to search in,  This is an optional param.  Dataobjects at all levels beneath this (not just the direct children) will be listed in the response. |
| Query Param | returnParent: If true, return the parent collection paths and metadata instead of the matched data object paths and metadata. Default is false. |
| 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,  }  **Note:**  **detailedResponse**: By default, search list returns only path of matched data objects. If “detailedResponse=true”, data objects 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 data objects 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. |

|  |  |
| --- | --- |
| Success Response | Example 1: - detailedResponse: false, returnParent: false  HTTP/1.1 200 OK  Content-Type: application/json  {  "dataObjectPaths": [    “/NCI\_DOE\_Archive/JDACS4C/JDACS4C\_Pilot\_2/KRAS4b\_Simulation\_Data/wt.pdb ",  /NCI\_DOE\_Archive/JDACS4C/JDACS4C\_Pilot\_2/KRAS4b\_Simulation\_Data/hvr3a\_pub.tar ",  “/NCI\_DOE\_Archive/JDACS4C/JDACS4C\_Pilot\_2/KRAS4b\_Simulation\_Data/hvr060-040\_pub.tar"  ],  "limit": 100,  "page": 1  }  **Example 2: - detailedResponse: true, returnParent: false**  {  "dataObjects": [{  "dataObject": {  "id": 6611663,  "collectionId": 6611544,  "dataName": null,  "collectionName": "/NCI\_DOE\_Archive/ATOM/Neurocrine\_H1\_Demonstration/H1\_Selectivity\_Assay",  "absolutePath": "/NCI\_DOE\_Archive/ATOM/Neurocrine\_H1\_Demonstration/H1\_Selectivity\_Assay/H1\_selectivity\_10\_29\_20.csv",  "dataSize": 0,  "dataPath": "/var/lib/irods/iRODS/Vault/home/NCI\_DOE\_Archive/ATOM/Neurocrine\_H1\_Demonstration/H1\_Selectivity\_Assay/H1\_selectivity\_10\_29\_20.csv",  "dataOwnerName": "ncidoesvcp2",  "createdAt": "2020-11-20T17:10:35.000+00:00"  },  "metadataEntries": {  "selfMetadataEntries": [{  "attribute": "source\_file\_size",  "value": "109651",  "unit": null,  "level": 1,  "levelLabel": "DataObject",  "collectionId": 6611544  }, {  "attribute": "data\_transfer\_completed",  "value": "11-20-2020 12:10:37",  "unit": null,  "level": 1,  "levelLabel": "DataObject",  "collectionId": 6611544  }, {  "attribute": "data\_transfer\_status",  "value": "ARCHIVED",  "unit": null,  "level": 1,  "levelLabel": "DataObject",  "collectionId": 6611544  },{  "attribute": "program\_description",  "value": "The Accelerating Therapeutics for Opportunities in Medicine (ATOM) Consortium is a public-private partnership whose mission is to transform drug discovery by accelerating the development of more effective therapies for patients. It is comprised of a partnership between the Frederick National Laboratory for Cancer Research, GSK, Lawrence Livermore National Laboratory, and the University of California, San Francisco.",  "unit": null,  "level": 4,  "levelLabel": "Program",  "collectionId": 6611544  }, {  "attribute": "configuration\_id",  "value": "4ac6a3d2-48bf-4a2e-8374-fe5449561d9b",  "unit": null,  "level": 4,  "levelLabel": "Program",  "collectionId": 6611544  }, {  "attribute": "program\_name",  "value": "Accelerating Therapeutics for Opportunities in Medicine (ATOM)",  "unit": null,  "level": 4,  "levelLabel": "Program",  "collectionId": 6611544  }]  },  "percentComplete": null,  "permission": null  }],  "dataObjectPaths": [],  "page": 1,  "limit": 100,  "totalCount": 2  }  NCI - 02133266 - ML: utils sehgalu2$  **Example 3: - detailedResponse: true, returnParent: true**  {  "collections": [{  "collection": {  "collectionId": 6611544,  "collectionName": "/NCI\_DOE\_Archive/ATOM/Neurocrine\_H1\_Demonstration/H1\_Selectivity\_Assay",  "absolutePath": "/NCI\_DOE\_Archive/ATOM/Neurocrine\_H1\_Demonstration/H1\_Selectivity\_Assay",  "collectionParentName": "/NCI\_DOE\_Archive/ATOM/Neurocrine\_H1\_Demonstration",  "collectionOwnerName": "ncidoesvcp2",  "collectionOwnerZone": "ncifprodZone",  "collectionMapId": "0",  "collectionInheritance": "1",  "comments": null,  "info1": null,  "info2": null,  "createdAt": "2020-11-20T17:06:25.000+00:00",  "specColType": null,  "subCollections": [],  "dataObjects": []  },  "metadataEntries": {  "selfMetadataEntries": [{  "attribute": "metadata\_updated",  "value": "01-29-2021 00:23:04",  "unit": null,  "level": 1,  "levelLabel": "Asset",  "collectionId": 6611544  }, {  "attribute": "asset\_identifier",  "value": "H1\_Selectivity\_Assay",  "unit": null,  "level": 1,  "levelLabel": "Asset",  "collectionId": 6611544  }, {  "attribute": "access\_group",  "value": "public",  "unit": null,  "level": 1,  "levelLabel": "Asset",  "collectionId": 6611544  }, {  "attribute": "asset\_name",  "value": "H1 Selectivity Assay",  "unit": null,  "level": 1,  "levelLabel": "Asset",  "collectionId": 6611544  }, {  "attribute": "asset\_type",  "value": "Dataset",  "unit": null,  "level": 1,  "levelLabel": "Asset",  "collectionId": 6611544  }, {  "attribute": "registered\_by\_name",  "value": "NCI\_DOE NCI\_DOE",  "unit": null,  "level": 1,  "levelLabel": "Asset",  "collectionId": 6611544  }, {  "attribute": "uuid",  "value": "e497fd73-7f50-465d-a6be-557df828eae1",  "unit": null,  "level": 1,  "levelLabel": "Asset",  "collectionId": 6611544  }, {  "attribute": "configuration\_id",  "value": "4ac6a3d2-48bf-4a2e-8374-fe5449561d9b",  "unit": null,  "level": 1,  "levelLabel": "Asset",  "collectionId": 6611544  }, {  "attribute": "collection\_type",  "value": "Asset",  "unit": null,  "level": 1,  "levelLabel": "Asset",  "collectionId": 6611544  }, {  "attribute": "description",  "value": "Histamine-1 (H1), Muscarinic Receptors 1-5 (M1-M5) and hERG binding affinity along with ligand structural data.",  "unit": null,  "level": 1,  "levelLabel": "Asset",  "collectionId": 6611544  },  {  "attribute": "registered\_by",  "value": "ncidoesvcp2",  "unit": null,  "level": 1,  "levelLabel": "Asset",  "collectionId": 6611544  }],  "parentMetadataEntries": [{  "attribute": "registered\_by\_name",  "value": "NCI\_DOE NCI\_DOE",  "unit": null,  "level": 2,  "levelLabel": "Study",  "collectionId": 6611412  }, {  "attribute": "metadata\_updated",  "value": "11-20-2020 12:01:58",  "unit": null,  "level": 2,  "levelLabel": "Study",  "collectionId": 6611412  }, {  "attribute": "configuration\_id",  "value": "4ac6a3d2-48bf-4a2e-8374-fe5449561d9b",  "unit": null,  "level": 2,  "levelLabel": "Study",  "collectionId": 6611412  }, {  "attribute": "uuid",  "value": "1ca4b087-5c76-4bc3-9354-2f3122500e45",  "unit": null,  "level": 2,  "levelLabel": "Study",  "collectionId": 6611412  }, {  "attribute": "collection\_type",  "value": "Study",  "unit": null,  "level": 2,  "levelLabel": "Study",  "collectionId": 6611412  }, {  "attribute": "description",  "value": "Design and synthesize selective H1 antagonists",  "unit": null,  "level": 2,  "levelLabel": "Study",  "collectionId": 6611412  }, {  "attribute": "study\_name",  "value": "Neurocrine H1 Demonstration",  "unit": null,  "level": 2,  "levelLabel": "Study",  "collectionId": 6611412  },  {  "attribute": "registered\_by",  "value": "ncidoesvcp2",  "unit": null,  "level": 2,  "levelLabel": "Study",  "collectionId": 6611412  }, {  "attribute": "study\_identifier",  "value": "Neurocrine\_H1\_Demonstration",  "unit": null,  "level": 2,  "levelLabel": "Study",  "collectionId": 6611412  }, {  "attribute": "access\_group",  "value": "public",  "unit": null,  "level": 2,  "levelLabel": "Study",  "collectionId": 6611412  }, {  "attribute": "registered\_by",  "value": "ncidoesvcp2",  "unit": null,  "level": 3,  "levelLabel": "Program",  "collectionId": 6282617  }, {  "attribute": "access\_group",  "value": "public",  "unit": null,  "level": 3,  "levelLabel": "Program",  "collectionId": 6282617  }, {  "attribute": "metadata\_updated",  "value": "11-10-2020 16:46:44",  "unit": null,  "level": 3,  "levelLabel": "Program",  "collectionId": 6282617  }, {  "attribute": "collection\_type",  "value": "Program",  "unit": null,  "level": 3,  "levelLabel": "Program",  "collectionId": 6282617  }, {  "attribute": "program\_name",  "value": "Accelerating Therapeutics for Opportunities in Medicine (ATOM)",  "unit": null,  "level": 3,  "levelLabel": "Program",  "collectionId": 6282617  }, {  "attribute": "program\_identifier",  "value": "ATOM",  "unit": null,  "level": 3,  "levelLabel": "Program",  "collectionId": 6282617  }, {  "attribute": "program\_description",  "value": "The Accelerating Therapeutics for Opportunities in Medicine (ATOM) Consortium is a public-private partnership whose mission is to transform drug discovery by accelerating the development of more effective therapies for patients. It is comprised of a partnership between the Frederick National Laboratory for Cancer Research, GSK, Lawrence Livermore National Laboratory, and the University of California, San Francisco.",  "unit": null,  "level": 3,  "levelLabel": "Program",  "collectionId": 6282617  }, {  "attribute": "uuid",  "value": "2eba0f8c-0a21-469f-b5e4-13bfcb43dc21",  "unit": null,  "level": 3,  "levelLabel": "Program",  "collectionId": 6282617  }, {  "attribute": "registered\_by\_name",  "value": "NCI\_DOE NCI\_DOE",  "unit": null,  "level": 3,  "levelLabel": "Program",  "collectionId": 6282617  }, {  "attribute": "configuration\_id",  "value": "4ac6a3d2-48bf-4a2e-8374-fe5449561d9b",  "unit": null,  "level": 3,  "levelLabel": "Program",  "collectionId": 6282617  }]  },  "permission": null  }],  "collectionPaths": [],  "page": 1,  "limit": 100,  "totalCount": 1  } |
| Error Response | **Data Management Error:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  JSON:  {  "errorType": " DATABASE\_ERROR",  "message": " Failed to get Collections: ",  "stackTrace": "…"  }  **Invalid query attributes:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  {"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 |

## Get Data File (V2)

|  |  |
| --- | --- |
| Title | Get Data File |
| Description | Get information about a specific data file including all of its hierarachical metadata |
| URL | /api/v2/dataObject/{urlPath} |
| Method | GET |
| Acceptable request representation | application/json  application/xml |
| Available response representation |  |
| URL Params | urlPath – The full path to the data file |
| Query params | includeAcl – If set to true, the API will provide the permission of the requesting user to this collection (READ, WRITE or OWN). This is an optional parameter and the default is false. |
| Data Params |  |
| Success Response | **HTTP/1.1 200 OK**  **Response when includeAcl=false**  {  "dataObjects": [{  "dataObject": {  "id": 6645957,  "collectionId": 6612116,  "dataName": null,  "collectionName": "/NCI\_DOE\_Archive/JDACS4C/JDACS4C\_Pilot\_2/KRAS4b\_Simulation\_Data",  "absolutePath": "/NCI\_DOE\_Archive/JDACS4C/JDACS4C\_Pilot\_2/KRAS4b\_Simulation\_Data/wt.pdb",  "dataSize": 0,  "dataPath": "/var/lib/irods/iRODS/Vault/home/NCI\_DOE\_Archive/JDACS4C/JDACS4C\_Pilot\_2/KRAS4b\_Simulation\_Data/wt.pdb",  "dataOwnerName": "ncifhpcdmsvcp",  "createdAt": "2020-12-02T20:04:45.000+00:00"  },  "metadataEntries": {  "selfMetadataEntries": [{  "attribute": "source\_file\_size",  "value": "10436626",  "unit": null,  "level": null,  "levelLabel": null,  "collectionId": null  }, {  "attribute": "checksum",  "value": "4f288f2b15c0630c5fca0692d0d66632",  "unit": null,  "level": null,  "levelLabel": null,  "collectionId": null  }, {  "attribute": "s3\_archive\_configuration\_id",  "value": "115249fe-9f1f-4139-a72e-43e8902f38f1",  "unit": null,  "level": null,  "levelLabel": null,  "collectionId": null  }, {  "attribute": "source\_file\_id",  "value": "1rVP1EbUC3R4rpf8uzIoyGoFeAgg0v-rW",  "unit": null,  "level": null,  "levelLabel": null,  "collectionId": null  }, {  "attribute": "program\_identifier",  "value": "JDACS4C",  "unit": "EMPTY\_ATTR\_UNIT",  "level": 4,  "levelLabel": "Program",  "collectionId": null  }, {  "attribute": "uuid",  "value": "b75a252a-1d99-4a33-acae-f7329a60870b",  "unit": "EMPTY\_ATTR\_UNIT",  "level": 4,  "levelLabel": "Program",  "collectionId": null  }, {  "attribute": "configuration\_id",  "value": "4ac6a3d2-48bf-4a2e-8374-fe5449561d9b",  "unit": "EMPTY\_ATTR\_UNIT",  "level": 4,  "levelLabel": "Program",  "collectionId": null  }, {  "attribute": "program\_name",  "value": "Advanced Computing Solutions for Cancer (JDACS4C)",  "unit": "EMPTY\_ATTR\_UNIT",  "level": 4,  "levelLabel": "Program",  "collectionId": null  }, {  "attribute": "program\_description",  "value": "The Joint Design of Advanced Computing Solutions for Cancer (JDACS4C) program was created as part of the Cancer Moonshot to accelerate cancer research using emerging exascale computing capabilities. It is a cross-agency collaboration between NCI and the Department of Energy (DOE). Investigators from NCI and the Frederick National Laboratory for Cancer Research work collaboratively with experts in computational, data, and physical sciences from four DOE national laboratories: Argonne, Los Alamos, Lawrence Livermore, and Oak Ridge.",  "unit": "EMPTY\_ATTR\_UNIT",  "level": 4,  "levelLabel": "Program",  "collectionId": null  }, {  "attribute": "access\_group",  "value": "public",  "unit": "EMPTY\_ATTR\_UNIT",  "level": 4,  "levelLabel": "Program",  "collectionId": null  }, {  "attribute": "metadata\_updated",  "value": "07-31-2020 17:27:10",  "unit": "EMPTY\_ATTR\_UNIT",  "level": 4,  "levelLabel": "Program",  "collectionId": null  }]  },  "percentComplete": null,  "permission": null  }],  "dataObjectPaths": [],  "page": null,  "limit": null,  "totalCount": null  }  **Response when includeAcl=true**  {  "dataObjects": [{  "dataObject": {  "id": 6645957,  "collectionId": 6612116,  "dataName": null,  "collectionName": "/NCI\_DOE\_Archive/JDACS4C/JDACS4C\_Pilot\_2/KRAS4b\_Simulation\_Data",  "absolutePath": "/NCI\_DOE\_Archive/JDACS4C/JDACS4C\_Pilot\_2/KRAS4b\_Simulation\_Data/wt.pdb",  "dataSize": 0,  "dataPath": "/var/lib/irods/iRODS/Vault/home/NCI\_DOE\_Archive/JDACS4C/JDACS4C\_Pilot\_2/KRAS4b\_Simulation\_Data/wt.pdb",  "dataOwnerName": "ncifhpcdmsvcp",  "createdAt": "2020-12-02T20:04:45.000+00:00"  },  "metadataEntries": {  "selfMetadataEntries": [{  "attribute": "source\_file\_size",  "value": "10436626",  "unit": null,  "level": null,  "levelLabel": null,  "collectionId": null  }, {  "attribute": "checksum",  "value": "4f288f2b15c0630c5fca0692d0d66632",  "unit": null,  "level": null,  "levelLabel": null,  "collectionId": null  }, {  "attribute": "configuration\_id",  "value": "4ac6a3d2-48bf-4a2e-8374-fe5449561d9b",  "unit": "EMPTY\_ATTR\_UNIT",  "level": 2,  "levelLabel": "Asset",  "collectionId": null  }, {  "attribute": "metadata\_updated",  "value": "01-28-2021 02:06:23",  "unit": "EMPTY\_ATTR\_UNIT",  "level": 2,  "levelLabel": "Asset",  "collectionId": null  }, {  "attribute": "uuid",  "value": "5b634039-3e36-400d-8647-0d46f7733f71",  "unit": "EMPTY\_ATTR\_UNIT",  "level": 2,  "levelLabel": "Asset",  "collectionId": null  }, {  "attribute": "asset\_name",  "value": "KRAS4b Simulation Data",  "unit": "EMPTY\_ATTR\_UNIT",  "level": 2,  "levelLabel": "Asset",  "collectionId": null  }, {  "attribute": "publication",  "value": "Goswami D, Chen D, Yang Y, Gudla RP, Columbus J, Worthy K, Rigby M, Wheeler M, Mukhopadhyay S, Powell K, Burgan W, Wall V, Esposito D, Simanshu D, Lightstone FC, Nissley DV, McCormick F, Turbyville T. 2020. Membrane interactions of the globular domain and the hypervariable region of KRAS4b define its unique diffusion behavior.",  "unit": "EMPTY\_ATTR\_UNIT",  "level": 2,  "levelLabel": "Asset",  "collectionId": null  }, {  "attribute": "metadata\_updated",  "value": "07-31-2020 17:27:10",  "unit": "EMPTY\_ATTR\_UNIT",  "level": 4,  "levelLabel": "Program",  "collectionId": null  }]  },  "percentComplete": null,  "permission": "OWN"  }],  "dataObjectPaths": [],  "page": null,  "limit": null,  "totalCount": null  }  NCI - 02133266 - ML: utils sehgalu2$ |
| 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 "  }  **Data file doesn’t exist:**  HTTP/1.1 204 No Content |

## Download Data File (v2)

|  |  |
| --- | --- |
| Title | Download/transfer data file from Archive storage to another location |
| Description | This API transfers a data file from archive storage to one of 4 possible destinations:   1. User’s file system. Synchronous downlowd 2. Globus endpoint. Asynchronous download 3. AWS S3. Asynchronous download 4. Google Drive Asynchronous download |
| URL | /api/v2/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 local file system synchronously  **JSON**:  {}  In case the data file is either a ZIP, TAR ot TGZ (tar.gz), on synchronous download, the API accepts an optional ‘filter’, so that only selected files from the ZIP/TAR/TGZ are included in the download  {  "synchronousDownloadFilter": {  "compressedArchiveType": "ZIP",  "includePatterns": ["\*\*/pi-xyz\*.\*", "some-folder/somefile" ]  }  }  Note the folder structure in the ZIP/TAR/TGZ that was archived is maintained in the filtered ZIP/TAR/TGZ that is returned with just the files asked for.  To request a download to a Globus endpoint  JSON:  {  "globusDownloadDestination": {  "destinationLocation": {  "fileContainerId": "4a3b132a-815f-11e7-8dff-22000b9923ef",  "fileId": "test-12-02-18"  },  "destinationOverwrite" : true,  }  }   * destinationLocation is the Globus endpoint * destinationOverwrite – This is an optional indicator to allow overwrite of file in Globus if it exists. If not provided, it is defaulted to false   **To request a download to an AWS S3**  **JSON:**  {  "s3DownloadDestination": {  "destinationLocation": {  "fileContainerId": "DOE-1",  "fileId": "demo-file-1-28-21"  },  "account" : {  "accessKey" : "\*\*\*",  "secretKey" : "\*\*\*",  "region" :"us-east-1",  }  }  }  destinationLocation is the S3 Bucket.  account is the S3 account  **To request a download to a Google Drive**  **JSON:**  {  "googleDriveDownloadDestination": {  "destinationLocation": {  "fileContainerId": "MyDrive",  "fileId": "gd-test-file-med-size-1-28-22"  },  "accessToken" : "<token>"  }} |
| Success Response | **Globus and S3download response**  HTTP/1.1 200 OK  Content-Type: application/json  {     "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  {  "errorType": "INVALID\_REQUEST\_INPUT",  "message": "Data object not found: <Path>",  "stackTrace": "gov.nih.nci.hpc.exception.HpcException: Data object not found:  }  **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  {  "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  {  "errorType": "DATA\_TRANSFER\_ERROR",  "message": "Failed to activate endpoint: nihcr#gridftp",    }  **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  {  "errorType": "INVALID\_REQUEST\_INPUT",  "message": "Invalid file location",  "stackTrace": "gov.nih.nci.hpc.exception.HpcException: Invalid file location[INVALID\_REQUEST\_INPUT]”  } |

## Download Collection (V2)

|  |  |
| --- | --- |
| Title | Download/transfer collection from Archive storage to another location |
| Description | This API transfers a collection from archive storage to another accessible Globus location, or AWS S3 bucket. Path represents the logical path of the collection registered with MoDaC. User should provide either a Globus destination which describes the endpoint, S3 destination which includes S3 bucket and account, or Google Drive destination which includes a drive, folder and access token  An optional ‘destinationOverwrite’ indicator can be provided for Globus destination, 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  Valid S3 account information is mandatory for S3 destination |
| URL | /api/v2/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**  {  "globusDownloadDestination": {  "destinationOverwrite" : true,  "destinationLocation": {  "fileContainerId": "4a3b132a-815f-11e7-8dff-22000b9923ef",  "fileId": "/colllection\_1\_29\_21"  }  }  }  **To request a download to AWS S3**  {  "s3DownloadDestination": {  "destinationLocation": {  "fileContainerId": "ccbr-sbg-1",  "fileId": "coll-many-files-12-10-18"  },  "account" : {  "accessKey" : "<insert-access-key",  "secretKey" : "<insert-secret-key",  "region" :"us-east-2"  }  }  }  **To request a download to Google Drive**  {  "googleDriveDownloadDestination": {  "destinationLocation": {  "fileContainerId": "MyDrive",  "fileId": "coll-download-test-01-28-21"  },  "accessToken" : "<token>"  }} |
| Success Response | **HTTP/1.1 200 OK**  Content-Type: application/json  {     "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  {  "errorType": "INVALID\_REQUEST\_INPUT",  "message": "Collection not found: <Path>",  "stackTrace": "gov.nih.nci.hpc.exception.HpcException: Data object not found:”  } |

## Download Collection List

|  |  |
| --- | --- |
| Title | Download/transfer collections list from Archive storage to another location |
| Description | This API transfers a list of collections from MoDaC to another accessible Globus location, AWS S3 bucket, or Google Drive. The caller provides a list of collection paths, and destination information which is the Globus endpoint to download to, the AWS S3 bucket and account, or Google Drive access token and folder name.  Note: Write access right needs to be granted to the service account if a user downloads a data file from Archive to a Globus end point. S3 account must be provided for S3 destination  An optional ‘appendPathToDownloadDestination’ indicator can be provided. If set to true, the entire object path (as it is in iRODS) will be used when files are created in the download destination, otherwise, just the file name from iRODs is used. By default this indicator is set to true. |
| URL | /api/v2/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**  {                  "collectionPaths" : [                      "/NCI\_DOE\_Archive/JDACS4C/JDACS4C\_Pilot\_2/KRAS4b\_Simulation\_Data",                      "/NCI\_DOE\_Archive/JDACS4C/JDACS4C\_Pilot\_2/KRAS4b\_Simulation\_Data\_1\_29"                  ],                  "globusDownloadDestination": {                                  "destinationOverwrite" : true,                                  "destinationLocation": {                                                  "fileContainerId": "4a3b132a-815f-11e7-8dff-22000b9923ef",                                                  "fileId": "bulk-collections-01-29-21"                                  }                  },  "appendPathToDownloadDestination" : true  }    **To request a download to AWS S3**  {                  "collectionPaths" : [                     "/NCI\_DOE\_Archive/JDACS4C/JDACS4C\_Pilot\_2/KRAS4b\_Simulation\_Data",                      "/NCI\_DOE\_Archive/JDACS4C/JDACS4C\_Pilot\_2/KRAS4b\_Simulation\_Data\_1\_29"                  ],                  "s3DownloadDestination": {                                  "destinationLocation": {                                                  "fileContainerId": "ccbr-sbg-1",                                                  "fileId": "eran-collection-bulk-download-08-17-19"                                  },                                  "account" : {                                                  "accessKey" : "",                                                  "secretKey" : "",                                                  "region" :"us-east-2"                                  }                  },  "appendPathToDownloadDestination" : true  }    **To request a download to Google Drive**  {                  "collectionPaths" : [  "/NCI\_DOE\_Archive/JDACS4C/JDACS4C\_Pilot\_2/KRAS4b\_Simulation\_Data",                      "/NCI\_DOE\_Archive/JDACS4C/JDACS4C\_Pilot\_2/KRAS4b\_Simulation\_Data\_1\_29"                  ],                 "googleDriveDownloadDestination": {  "destinationLocation": {  "fileContainerId": "MyDrive",  "fileId": "bulk-download-collections-05-24-20"  },  "accessToken" : "<Token>"  },  "appendPathToDownloadDestination" : true  } |
| Success Response | HTTP/1.1 200 OK  Content-Type: application/json  {  "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 collection path is given  HTTP/1.1 400 Bad Request  Content-Type: application/json  {  "errorType": "INVALID\_REQUEST\_INPUT",  "message": "Collection doesn't exist: <Path>",  } |

## Download Data Object List (V2)

|  |  |
| --- | --- |
| 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, AWS S3 bucket, or Google Drive folder. The caller provides a list of data object paths, and destination information which is the Globus endpoint to download to, the AWS S3 bucket and account, or Google Drive access token and folder name  An optional ‘destinationOverwrite’ indicator can be provided for Globus destination, 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. S3 account must be provided for S3 destination  An optional ‘appendPathToDownloadDestination’ indicator can be provided. If set to true, the entire object path (as it is in iRODS) will be used when files are created in the download destination, otherwise, just the file name from iRODs is used. By default this indicator is set to true. |
| URL | /api/v2/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": [  "/NCI\_DOE\_Archive/JDACS4C/JDACS4C\_Pilot\_2/KRAS4b\_Simulation\_Data/wt.pdb”  "/NCI\_DOE\_Archive/JDACS4C/JDACS4C\_Pilot\_2/KRAS4b\_Simulation\_Data/hvr060-040\_pub.tar”  “/NCI\_DOE\_Archive/JDACS4C/JDACS4C\_Pilot\_2/KRAS4b\_Simulation\_Data/hvr070-030\_pub.tar”  ],  "globusDownloadDestination": {  "destinationOverwrite": true,  "destinationLocation": {  "fileContainerId": "16572124-19cb-11e9-934d-0e3d676669f4",  "fileId": "bulk-dataobjects\_1\_28\_2021"  }  },  "appendPathToDownloadDestination": true  }  **To request a download to an AWS S3**  **{**  "dataObjectPaths" : [  "/NCI\_DOE\_Archive/JDACS4C/JDACS4C\_Pilot\_2/KRAS4b\_Simulation\_Data/wt.pdb”  "/NCI\_DOE\_Archive/JDACS4C/JDACS4C\_Pilot\_2/KRAS4b\_Simulation\_Data/hvr060-040\_pub.tar”  “/NCI\_DOE\_Archive/JDACS4C/JDACS4C\_Pilot\_2/KRAS4b\_Simulation\_Data/hvr070-030\_pub.tar”  ],  "s3DownloadDestination": {  "destinationLocation": {  "fileContainerId": "DOE\_bucket",  "fileId": "bulk-1-29-21"  },  "account" : {  "accessKey" : "<insert-access-key>",  "secretKey" : "<insert-secret-key>",  "region" :"us-east-2"  }  },  "appendPathToDownloadDestination" : true  }  **To request a download to a Google Drive**  **{**  "dataObjectPaths" : [  "/NCI\_DOE\_Archive/JDACS4C/JDACS4C\_Pilot\_2/KRAS4b\_Simulation\_Data/wt.pdb”  "/NCI\_DOE\_Archive/JDACS4C/JDACS4C\_Pilot\_2/KRAS4b\_Simulation\_Data/hvr060-040\_pub.tar”  “/NCI\_DOE\_Archive/JDACS4C/JDACS4C\_Pilot\_2/KRAS4b\_Simulation\_Data/hvr070-030\_pub.tar”  ],  "googleDriveDownloadDestination": {  "destinationLocation": {  "fileContainerId": "MyDrive",  "fileId": "bulk-download-01-29-21"  },  "accessToken" : "<Token>"  },  "appendPathToDownloadDestination" : true  } |
| Success Response | HTTP/1.1 200 OK  Content-Type: application/json  {  "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  {  "errorType": "INVALID\_REQUEST\_INPUT",  "message": "Data Object not found: <Path>",  } |

## Register Data File (V2)

|  |  |
| --- | --- |
| Title | Register data file |
| Description | Register data file along with its metadata into MoDaC. Registering a data file requires a registrar and PI to have a valid account with the upload permission on MoDaC. Registering a data file would transfer dataset from its original endpoint or file system to a target archive storage.  Registering data file supports seven data transfer options:   1. Synchronous data transfer from local file system to MoDaC storage. 2. Asynchronous data transfer from a Globus endpoint to MoDaC storage. MoDaC uses service account to transfer data from your source location. You need to share data with “MODAC-PROD-App-Accts-Pool-NCI\_DOE“ before submitting your request. 3. User upload data using a pre-signed URL generated by the system. The user can request a single-part upload, i.e. upload the file w/ one URL, or ask for multipart uploads, specifying the number of parts. In this case the response will include a URL for each part upload, and a multipart request ID that is needed to call the ‘complete multipart upload’ API. 4. Asynchronous data transfer from AWS S3. User’s needs to provide S3 account information to be used to perform the transfer. 5. Registering w/ a link. In this case, no upload is performed, but rather the new data object is linked to another. 6. Asynchronous data transfer from Google Drive. User’s needs to provide Google Drive access token to be used to perform the transfer. 7. Asynchronous data transfer from the server file system (NAS).   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) (Only for sync upload)    1. ContentType: application/octet-stream   ContentID: dataObject    If you want the system to generate a pre-signed URL, then “dataObjectRegistration” JSON should include ‘generateUploadRequestURL’ element set to true.  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 and higher-level collections 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 provided through parentCollectionBulkMetadataEntires. In this case the entire collection hierarchy above the registered collection will be created. If part or all of the parent and higher-level collection exist, metadata can be updated/added with the list of metadata provided for the parent collection.  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.  For synchronous registration, caller can specify an automatic metadata extraction from the attached file. The server will parser the file and extract its metadata. By default, no metadata extraction is performed.  The caller can optionally attach a list of metadata extracted from the physical file – These metadata should be provided in a separate list, so they can be separated from the metadata defined by the caller. |
| URL | /api/v2/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 | **Synchronous Upload Example**  {  "callerObjectId": "<user-defined-base-path-in-archive (optional) >",  "checksum": "<data-checksum (optional) >",  "extractMetadata": false,  "metadataEntries": [{  "attribute": "description",  "value": "Dataset"  }, {  "attribute": "Asset Type",  "value": "test\_Asset"  }, {  "attribute": "file\_type",  "value": "Tensorflow Graphdef"  }, {  "attribute": "Asset\_identifier",  "value": "Asset\_Udit"  }, {  "attribute": "Asset\_name",  "value": "Sehgalu2"  }]  }  Note: in sync upload, you also need to attach the data file itself (in addition to JSON).  Asynchronous Upload w/ Globus Example  {{  "globusUploadSource": {  "sourceLocation": {  "fileContainerId": "16572124-19cb-11e9-934d-0e3d676669f4",  "fileId": "MoDaC\_file\_1\_29\_2021"  }  },  "metadataEntries": [{  "attribute": "description",  "value": "Dataset"  }, {  "attribute": "Asset Type",  "value": "test\_Asset"  }, {  "attribute": "file\_type",  "value": "Tensorflow Graphdef"  }, {  "attribute": "Asset\_identifier",  "value": "Asset\_Udit"  }, {  "attribute": "Asset\_name",  "value": "Sehgalu2"  }],  "extractedMetadataEntries": [{  "attribute": "extracted-name",  "value": "extracted-value"  }],  "createParentCollections": "true",  "parentCollectionsBulkMetadataEntries": {  "pathsMetadataEntries": [{  "path": "/NCI\_DOE\_Archive/JDACS4C/JDACS4C\_Pilot\_2/KRAS4b\_Simulation\_Data",  "pathMetadataEntries": [{  "attribute": "collection\_type",  "value": "Asset"  },  {  "attribute": "run\_name",  "value": "124533\_NS500417\_0125\_AUTJXY"  },  {  "attribute": "run\_date",  "value": "05-23-20"  },  {  "attribute": "sequencing\_platform",  "value": "ATAC-Seq"  },  {  "attribute": "sequencing\_application\_type",  "value": "NextSeq"  },  {  "attribute": "read\_length",  "value": "R1: 75, R2: 75"  },  {  "attribute": "pooling",  "value": "n=4"  }  ]  }]  }  }  Generate Upload URL (single-part) Example  {  "generateUploadRequestURL": true,  "callerObjectId": "<user-defined-base-path-in-archive (optional) >",  "metadataEntries": [{  "attribute": "description",  "value": "Dataset"  }, {  "attribute": "Asset Type",  "value": "test\_Asset"  }, {  "attribute": "file\_type",  "value": "Tensorflow Graphdef"  }, {  "attribute": "Asset\_identifier",  "value": "Asset\_Udit"  }, {  "attribute": "Asset\_name",  "value": "Sehgalu2"  }]  }  Generate Upload URLs (multi-part) Example  {  "generateUploadRequestURL": true,  "uploadParts": true,  "callerObjectId": "<user-defined-base-path-in-archive (optional) >",  "metadataEntries": [{  "attribute": "name",  "value": "Set100"  }]  }  **Asynchronous Upload from AWS S3 Example**  {{  "s3UploadSource": {  "sourceLocation": {  "fileContainerId": "<S3-bucket-name>",  "fileId": "<S3-object-key>"  },  "account": {  "accessKey": "<aws-access-key>",  "secretKey": "<aws-secret-key>",  "region": "<aws-region>"  }  },  "callerObjectId": "<user-defined-base-path-in-archive (optional) >",  "metadataEntries": [{  "attribute": "description",  "value": "Dataset"  }, {  "attribute": "Asset Type",  "value": "test\_Asset"  }, {  "attribute": "file\_type",  "value": "Tensorflow Graphdef"  }, {  "attribute": "Asset\_identifier",  "value": "Asset\_Udit"  }, {  "attribute": "Asset\_name",  "value": "Sehgalu2"  }]  }  **Asynchronous Upload from Google Drive Example**  {  "googleDriveUploadSource": {  "sourceLocation": {  "fileContainerId": "MyDrive",  "fileId": "/MoDaC/Test-data-file"  },  "accessToken": "<token> "  },  "callerObjectId": "<user-defined-base-path-in-archive (optional) >",  "metadataEntries": [{  "attribute": "description",  "value": "Dataset"  }, {  "attribute": "Asset Type",  "value": "test\_Asset"  }, {  "attribute": "file\_type",  "value": "Tensorflow Graphdef"  }, {  "attribute": "Asset\_identifier",  "value": "Asset\_Udit"  }, {  "attribute": "Asset\_name",  "value": "Sehgalu2"  }]  }  **Asynchronous Upload from Server(NAS)**  {  "fileSystemUploadSource": {  "sourceLocation": {  "fileContainerId": "nas-file-system-name",  "fileId": "/home/folder/data-object"  }  },  "metadataEntries": [{  "attribute": "description",  "value": "Dataset"  }, {  "attribute": "Asset Type",  "value": "test\_Asset"  }, {  "attribute": "file\_type",  "value": "Tensorflow Graphdef"  }, {  "attribute": "Asset\_identifier",  "value": "Asset\_Udit"  }, {  "attribute": "Asset\_name",  "value": "Sehgalu2"  }]  } |
| Success Response | HTTP/1.1 201 Created  Content-Length: 0  Note: If a presigned upload URL was requested (single/multi part), a JSON/XML will be additionally returned:  **Single-part Upload Response:**  {  “uploadRequestURL” : “https://…..”  }  **Multi-part Upload Response:**  {  "multipartUpload": {  "id": "MDA3Yzk2NzYxNTg1NzUyMTA2MjEw",  "parts": [{  "partNumber": 1,  "partUploadRequestURL": "http://..."  },  {  "partNumber": 2,  "partUploadRequestURL": "http://..."  },  {  "partNumber": 3,  "partUploadRequestURL": "http://..."  }  ]  }  } |
| Error Response | **Data file path already exists:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  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  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  JSON:  {  "errorType": "REQUEST\_REJECTED",  "requestRejectReason": "INVALID\_DATA\_OBJECT\_PATH",  "message": "Invalid data object path. Directory doesn't exist:  "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  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  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  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  JSON:  {  "errorType": "INVALID\_REQUEST\_INPUT",  "message": "Invalid Data Transfer Input",  "stackTrace": "…"  }  **Invalid Request Input:**  HTTP/1.1 400 Bad Request  Content-Type: application/json  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  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  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": "… "  } |

## Generate Download Request URL

|  |  |
| --- | --- |
| Title | Generate a (pre-signed) download request URL |
| Description | This API generates a URL to be used to download a data object file directly from the archive. This API is supported by Cleversafe archive only. |
| URL | /api/dataObject/{path}/generateDownloadRequestURL |
| Method | POST |
| Acceptable request representation | application/json  application/xml |
| Available response representation |  |
| URL Params | path - The data object path to generate the download URL for |
| Data Params |  |
| Success Response | **HTTP/1.1 200 OK**  Content-Type: application/json  {  "downloadRequestURL": "http://nciarchive.nci.nih.gov/... ",  "dataTransferType": "S\_3"  }  Note: The user can then copy/paste the generated URL into the browser and it will download it, or use wget command to download it to a local file. |
| Error Response | **POSIX Archive:**  This error is thrown if a download URL is requested for POSIX archive  HTTP/1.1 400 Bad Request  {  "errorType": "INVALID\_REQUEST\_INPUT",  "message": "S3 download request is not supported for POSIX based file system archive"  } |