Data Repository Service

Table of Contents

Chapter 1. Overview

https://github.com/ga4gh/data-repository-service-schemas

1.1. Version information

Version : 0.6.0

1.2. Contact information

Contact : GA4GH Cloud Work Stream
Contact Email : ga4gh-cloud@ga4gh.org

1.3. License information

License : Apache 2.0
License URL : https://raw.githubusercontent.com/ga4gh/data-repository-service-schemas/master/LICENSE
Terms of service : null

1.4. URI scheme

BasePath : /ga4gh/drs/v1
Schemes : HTTPS, HTTP

1.5. Consumes

- `application/json`

1.6. Produces

- `application/json`

Chapter 2. Executive Summary

The Data Repository Service (DRS) API provides a generic API on top of existing object storage systems so workflow systems can access data in a single, standard way regardless of where it's stored.

Key features of the API:

- **Data Object management**, which enables the creation, updating, deletion, versioning, and unique identification of files and data bundles (flat collections of files)
- **Data Object querying**, which can locate data objects across different cloud environments and DRS implementations.

Chapter 3. Introduction

This document describes the DRS API and provides details on the specific endpoints, request formats, and response. It is intended to provide key information for developers of DRS-compatible services as well as clients that will call these DRS services.

Use cases include:

- GET of resource ID and return JSON metadata describing object
- GET of resource ID and return JSON with URLs for data read access.
- GET of resources by alias
- Bundles of multiple objects
- Versioning of objects (and bundles if supported)

Chapter 4. Standards

The DRS API specification is written in OpenAPI and embodies a RESTful service philosophy. It uses JSON in requests and responses and standard HTTP/HTTPS for information transport.

Chapter 5. Authorization & Authentication

Users must supply credentials that establish their identity and authorization in order to use a DRS endpoint. We recommend that DRS implementations use an OAuth2 bearer token, although they can choose other mechanisms if appropriate. DRS callers can use the `auth_instructions_url` from the service-info endpoint to learn how to obtain and use a bearer token for a particular implementation.

The DRS implementation is responsible for checking that a user is authorized to submit workflow run requests. The particular authorization policy is up to the DRS implementer.

Systems like DRS need to also address the ability to pass credentials with jobs for input and output access. In the current
version of DRS, the passing of credentials to authenticate and authorize access to inputs and outputs, as well as mandates about necessary file transfer protocols to support, are out of scope. However, parallel work on the Data Object Service is addressing ways to pass around access credentials with data object references, opening up the possibility that a future version of DRS will provide concrete mechanisms for workflow runs to access data using credentials different than those used for DRS. This is a work in progress and support of DOS in DRS will be added in a future release of DRS.

Chapter 6. Paths

6.1. Retrieve a Data Bundle

```
GET /bundles/{bundle_id}
```

6.1.1. Parameters

Type	Name	Schema
Path	**bundle_id** *required*	string

6.1.2. Responses

HTTP Code	Description	Schema
200	Successfully found the Data Bundle.	GetBundleResponse
400	The request is malformed.	ErrorResponse
401	The request is unauthorized.	ErrorResponse
403	The requester is not authorized to perform this action.	ErrorResponse
404	The requested Data Bundle wasn't found.	ErrorResponse
500	An unexpected error occurred.	ErrorResponse

6.1.3. Tags

- DataRepositoryService

6.2. Retrieve a Data Object

```
GET /objects/{object_id}
```

6.2.1. Parameters

Type	Name	Schema
Path	**object_id** *required*	string

6.2.2. Responses

HTTP Code	Description	Schema
200	The Data Object was found successfully.	GetObjectResponse
400	The request is malformed.	ErrorResponse
401	The request is unauthorized.	ErrorResponse
403	The requester is not authorized to perform this action.	ErrorResponse
404	The requested Data Object wasn't found	ErrorResponse
500	An unexpected error occurred.	ErrorResponse

6.2.3. Tags

- DataRepositoryService

6.3. Returns service version and other information

```
GET /service-info
```

6.3.1. Responses

HTTP Code	Description	Schema
200	Service information returned successfully	ServiceInfoResponse

6.3.2. Tags

- DataRepositoryService

Chapter 7. Definitions

7.1. AuthorizationMetadata

OPTIONAL
A set of key-value pairs that represent sufficient metadata to be granted access to a resource. It may be helpful to provide details about a specific provider, for example.

Name	Description	Schema
auth_type *optional*	The auth standard being used to make data available. For example, 'OAuth2.0'.	string
auth_url *optional*	The URL where the auth service is located, for example, a URL to get an OAuth token.	string

7.2. Bundle

Name	Description	Schema
aliases *optional*	A list of strings that can be used to identify this Data Bundle.	< string > array
checksums *required*	At least one checksum must be provided. The Data Bundle checksum is computed over all the checksums of the Data Objects that bundle contains.	< Checksum > array
created *required*	Timestamp of object creation in RFC3339.	string (date-time)
description *optional*	A human readable description.	string
id *required*	An identifier, unique to this Data Bundle	string
object_ids *required*	The list of Data Objects that this Data Bundle contains.	< string > array
system_metadata *optional*		SystemMetadata
updated *required*	Timestamp of update in RFC3339, identical to create timestamp in systems that do not support updates.	string (date-time)
user_metadata *optional*		UserMetadata

Name	Description	Schema
version *required*	A string representing a version, some systems may use checksum, a RFC3339 timestamp, or incrementing version number. For systems that do not support versioning please use your update timestamp as your version.	string

7.3. Checksum

Name	Description	Schema
checksum *required*	The hex-string encoded checksum for the Data.	string
type *optional*	The digest method used to create the checksum. If left unspecified md5 will be assumed. possible values: md5 # most blob stores provide a checksum using this multipart-md5 # multipart uploads provide a specialized tag in S3 sha256 sha512	string

7.4. ErrorResponse

An object that can optionally include information about the error.

Name	Description	Schema
msg *optional*	A detailed error message.	string
status_code *optional*	The integer representing the HTTP status code (e.g. 200, 404).	integer

7.5. GetBundleResponse

Name	Schema
bundle *optional*	Bundle

7.6. GetObjectResponse

Name	Schema
object *required*	Object

7.7. Object

Name	Description	Schema
aliases *optional*	A list of strings that can be used to find this Data Object. These aliases can be used to represent the Data Object's location in a directory (e.g. "bucket/folder/file.name") to make Data Objects more discoverable. They might also be used to represent	< string > array
checksums *required*	The checksum of the Data Object. At least one checksum must be provided.	< Checksum > array
created *required*	Timestamp of object creation in RFC3339.	string (date-time)
description *optional*	A human readable description of the contents of the Data Object.	string
id *required*	An identifier unique to this Data Object.	string
mime_type *optional*	A string providing the mime-type of the Data Object. For example, "application/json".	string
name *optional*	A string that can be optionally used to name a Data Object.	string
size *required*	The computed size in bytes.	string (int64)
updated *optional*	Timestamp of update in RFC3339, identical to create timestamp in systems that do not support updates.	string (date-time)
urls *optional*	The list of URLs that can be used to access the Data Object.	< URL > array
version *optional*	A string representing a version.	string

7.8. ServiceInfoResponse

Placeholder for the Info Object

Name	Description	Schema
contact *optional*	Maintainer contact info	object
description *optional*	Service description	string
license *optional*	License information for the exposed API	object
title *optional*	Service name	string

Name	Description	Schema
version *required*	Service version	string

7.9. SystemMetadata

OPTIONAL
These values are reported by the underlying object store.
A set of key-value pairs that represent system metadata about the object.

Type : object

7.10. URL

Name	Description	Schema
authorization_metadata *optional*		AuthorizationMetadata
system_metadata *optional*		SystemMetadata
url *required*	A URL that can be used to access the file.	string
user_metadata *optional*		UserMetadata

7.11. UserMetadata

OPTIONAL
A set of key-value pairs that represent metadata provided by the uploader.

Type : object