GA4GH Tool Discovery API

Table of Contents

Chapter 1. Overview

Proposed API for GA4GH (Global Alliance for Genomics & Health) tool repositories. A tool consists of a set of container images that are paired with a set of documents. Examples of documents include CWL (Common Workflow Language) or WDL (Workflow Description Language) or NFL (Nextflow) that describe how to use those images and a set of specifications for those images (examples are Dockerfiles or Singularity recipes) that describe how to reproduce those images in the future. We use the following terminology, a "container image" describes a container as stored at rest on a filesystem, a "tool" describes one of the triples as described above. In practice, examples of "tools" include CWL CommandLineTools, CWL Workflows, WDL workflows, and Nextflow workflows that reference containers in formats such as Docker or Singularity.

1.1. Version information

Version : 2.0.0

1.2. URI scheme

BasePath : /ga4gh/trs/v2

1.3. Tags

- GA4GH : A group of web resources proposed as a common standard for tool repositories

1.4. Produces

- `application/json`
- `text/plain`

1.5. External Docs

Description : Description of GA4GH Tool Registry (Exchange) Schema
URL : https://github.com/ga4gh/tool-registry-schemas

Chapter 2. Paths

2.1. List all tool types

```
GET /toolClasses
```

2.1.1. Description

This endpoint returns all tool-classes available.

2.1.2. Responses

HTTP Code	Description	Schema
200	A list of potential tool classes.	< ToolClass > array

2.1.3. Tags

- GA4GH

2.1.4. Security

Type	Name
apiKey	**BEARER**

2.2. List all tools

```
GET /tools
```

2.2.1. Description

This endpoint returns all tools available or a filtered subset using metadata query parameters.

2.2.2. Parameters

Type	Name	Description	Schema	Default
Query	**alias** *optional*	Support for this parameter is optional for tool registries that support aliases. If provided will only return entries with the given alias.	string	

Type	Name	Description	Schema	Default
Query	**author** *optional*	The author of the tool (TODO a thought occurs, are we assuming that the author of the CWL and the image are the same?).	string	
Query	**checker** *optional*	Return only checker workflows.	boolean	
Query	**description** *optional*	The description of the tool.	string	
Query	**id** *optional*	A unique identifier of the tool, scoped to this registry, for example 123456.	string	
Query	**limit** *optional*	Amount of records to return in a given page.	integer (int32)	1000
Query	**name** *optional*	The name of the image.	string	
Query	**offset** *optional*	Start index of paging. Pagination results can be based on numbers or other values chosen by the registry implementor (for example, SHA values). If this exceeds the current result set return an empty set. If not specified in the request, this will start at the beginning of the results.	string	
Query	**organization** *optional*	The organization in the registry that published the image.	string	
Query	**registry** *optional*	The image registry that contains the image.	string	
Query	**toolClass** *optional*	Filter tools by the name of the subclass (#/definitions/ToolClass)	string	
Query	**toolname** *optional*	The name of the tool.	string	

2.2.3. Responses

HTTP Code	Description	Schema
200	An array of Tools that match the filter. **Headers** : `next_page` (string) : A URL that can be used to reach the next page based on the current offset and page record limit. `last_page` (string) : A URL that can be used to reach the last page based on the current page record limit. `self_link` (string) : A URL that can be used to return to the current page later. `current_offset` (string) : The current start index of the paging used for this result. `current_limit` (integer) : The current page record limit used for this result.	< Tool > array

2.2.4. Tags

- GA4GH

2.2.5. Security

Type	Name
apiKey	**BEARER**

2.3. List one specific tool, acts as an anchor for self references

```
GET /tools/{id}
```

2.3.1. Description

This endpoint returns one specific tool (which has ToolVersions nested inside it).

2.3.2. Parameters

Type	Name	Description	Schema
Path	**id** *required*	A unique identifier of the tool, scoped to this registry, for example 123456.	string

2.3.3. Responses

HTTP Code	Description	Schema
200	A tool.	Tool
404	The tool can not be found.	Error

2.3.4. Tags

- GA4GH

2.3.5. Security

Type	Name
apiKey	**BEARER**

2.4. List versions of a tool

```
GET /tools/{id}/versions
```

2.4.1. Description

Returns all versions of the specified tool.

2.4.2. Parameters

Type	Name	Description	Schema
Path	**id** *required*	A unique identifier of the tool, scoped to this registry, for example 123456.	string

2.4.3. Responses

HTTP Code	Description	Schema
200	An array of tool versions.	< ToolVersion > array

2.4.4. Tags

- GA4GH

2.4.5. Security

Type	Name
apiKey	BEARER

2.5. List one specific tool version, acts as an anchor for self references

```
GET /tools/{id}/versions/{version_id}
```

2.5.1. Description

This endpoint returns one specific tool version.

2.5.2. Parameters

Type	Name	Description	Schema
Path	**id** *required*	A unique identifier of the tool, scoped to this registry, for example 123456.	string
Path	**version_id** *required*	An identifier of the tool version, scoped to this registry, for example v1. We recommend that versions use semantic versioning https://semver.org/spec/v2.0.0.html (For example, 1.0.0 instead of develop)	string

2.5.3. Responses

HTTP Code	Description	Schema
200	A tool version.	ToolVersion
404	The tool can not be found.	Error

2.5.4. Tags

- GA4GH

2.5.5. Security

Type	Name
apiKey	BEARER

2.6. Get the container specification(s) for the specified image.

```
GET /tools/{id}/versions/{version_id}/containerfile
```

2.6.1. Description

Returns the container specifications(s) for the specified image. For example, a CWL CommandlineTool can be associated with one specification for a container, a CWL Workflow can be associated with multiple specifications for containers.

2.6.2. Parameters

Type	Name	Description	Schema
Path	**id** *required*	A unique identifier of the tool, scoped to this registry, for example 123456.	string
Path	**version_id** *required*	An identifier of the tool version for this particular tool registry, for example v1.	string

2.6.3. Responses

HTTP Code	Description	Schema
200	The tool payload.	< FileWrapper > array
404	There are no container specifications for this tool.	Error

2.6.4. Tags

- GA4GH

2.6.5. Security

Type	Name
apiKey	BEARER

2.7. Get the tool descriptor for the specified tool

```
GET /tools/{id}/versions/{version_id}/{type}/descriptor
```

2.7.1. Description

Returns the descriptor for the specified tool (examples include CWL, WDL, or Nextflow documents).

2.7.2. Parameters

Type	Name	Description	Schema
Path	**id** *required*	A unique identifier of the tool, scoped to this registry, for example 123456.	string
Path	**type** *required*	The output type of the descriptor. Plain types return the bare descriptor while the "non-plain" types return a descriptor wrapped with metadata. Allowable values include "CWL", "WDL", "NFL", "PLAIN_CWL", "PLAIN_WDL", "PLAIN_NFL".	string
Path	**version_id** *required*	An identifier of the tool version, scoped to this registry, for example v1.	string

2.7.3. Responses

HTTP Code	Description	Schema
200	The tool descriptor.	FileWrapper
404	The tool descriptor can not be found.	Error

2.7.4. Tags

- GA4GH

2.7.5. Security

Type	Name
apiKey	BEARER

2.8. Get additional tool descriptor files relative to the main file

```
GET /tools/{id}/versions/{version_id}/{type}/descriptor/{relative_path}
```

2.8.1. Description

Descriptors can often include imports that refer to additional descriptors. This returns additional descriptors for the specified tool in the same or other directories that can be reached as a relative path. This endpoint can be useful for workflow engine implementations like cwltool to programmatically download all the descriptors for a tool and run it. This can optionally include other files described with FileWrappers such as test parameters and containerfiles.

2.8.2. Parameters

Type	Name	Description	Schema
Path	**id** *required*	A unique identifier of the tool, scoped to this registry, for example 123456.	string
Path	**relative_path** *required*	A relative path to the additional file (same directory or subdirectories), for example 'foo.cwl' would return a 'foo.cwl' from the same directory as the main descriptor. 'nestedDirectory/foo.cwl' would return the file from a nested subdirectory. Unencoded paths such 'sampleDirectory/foo.cwl' should also be allowed.	string
Path	**type** *required*	The output type of the descriptor. If not specified, it is up to the underlying implementation to determine which output type to return. Plain types return the bare descriptor while the "non-plain" types return a descriptor wrapped with metadata. Allowable values are "CWL", "WDL", "NFL", "PLAIN_CWL", "PLAIN_WDL", "PLAIN_NFL".	string
Path	**version_id** *required*	An identifier of the tool version for this particular tool registry, for example v1.	string

2.8.3. Responses

HTTP Code	Description	Schema
200	The tool descriptor.	FileWrapper
404	The tool can not be output in the specified type.	Error

2.8.4. Tags

- GA4GH

2.8.5. Security

Type	Name
apiKey	BEARER

2.9. Get a list of objects that contain the relative path and file type

```
GET /tools/{id}/versions/{version_id}/{type}/files
```

2.9.1. Description

Get a list of objects that contain the relative path and file type. The descriptors are intended for use with the /tools/{id}/versions/{version_id}/{type}/descriptor/{relative_path} endpoint. Returns a zip file of all files when format=zip is specified.

2.9.2. Parameters

Type	Name	Description	Schema
Path	**id** *required*	A unique identifier of the tool, scoped to this registry, for example 123456.	string
Path	**type** *required*	The output type of the descriptor. Examples of allowable values are "CWL", "WDL", and "NFL".	string
Path	**version_id** *required*	An identifier of the tool version for this particular tool registry, for example v1.	string
Query	**format** *optional*	Returns a zip file of all files when format=zip is specified.	enum (zip)

2.9.3. Responses

HTTP Code	Description	Schema
200	The array of File JSON responses.	< ToolFile > array
404	The tool can not be output in the specified type.	Error

2.9.4. Produces

- application/json
- application/zip

2.9.5. Tags

- GA4GH

2.9.6. Security

Type	Name
apiKey	**BEARER**

2.10. Get a list of test JSONs

```
GET /tools/{id}/versions/{version_id}/{type}/tests
```

2.10.1. Description

Get a list of test JSONs (these allow you to execute the tool successfully) suitable for use with this descriptor type.

2.10.2. Parameters

Type	Name	Description	Schema
Path	**id** *required*	A unique identifier of the tool, scoped to this registry, for example 123456.	string

Type	Name	Description	Schema
Path	**type** *required*	The type of the underlying descriptor. Allowable values include "CWL", "WDL", "NFL", "PLAIN_CWL", "PLAIN_WDL", "PLAIN_NFL". For example, "CWL" would return an list of ToolTests objects while "PLAIN_CWL" would return a bare JSON list with the content of the tests.	string
Path	**version_id** *required*	An identifier of the tool version for this particular tool registry, for example v1.	string

2.10.3. Responses

HTTP Code	Description	Schema
200	The tool test JSON response.	< FileWrapper > array
404	The tool can not be output in the specified type.	Error

2.10.4. Tags

- GA4GH

2.10.5. Security

Type	Name
apiKey	**BEARER**

Chapter 3. Definitions

3.1. Checksum

Name	Description	Schema
checksum *required*	The hex-string encoded checksum for the data.	string
type *required*	The digest method used to create the checksum. The value (e.g. `sha-256`) SHOULD be listed as `Hash Name String` in the GA4GH Checksum Hash Algorithm Registry. Other values MAY be used, as long as implementors are aware of the issues discussed in RFC6920. GA4GH may provide more explicit guidance for use of non-IANA-registered algorithms in the future.	string

3.2. DescriptorType

The type of descriptor that represents this version of the tool (e.g. CWL, WDL, or NFL). Note that these files can also include associated Docker/container files and test parameters that further describe a version of a tool.

Type : enum (CWL, WDL, NFL)

3.3. Error

Name	Description	Schema
code *required*	**Default** : `500`	integer (int32)
message *optional*	**Default** : `"Internal Server Error"`	string

3.4. FileWrapper

A file provides content for one of
- A tool descriptor is a metadata document that describes one or more tools.
- A tool document that describes how to test with one or more sample test JSON.
- A containerfile is a document that describes how to build a particular container image. Examples include Dockerfiles for creating Docker images and Singularity recipes for Singularity images

Name	Description	Schema
checksum *optional*	A production (immutable) tool version is required to have a hashcode. Not required otherwise, but might be useful to detect changes. **Example** : [{ "checksum" : "ea2a5db69bd20a42976838790bc29294df3af02b", "type" : "sha1" }]	< Checksum > array
content *optional*	The content of the file itself. One of url or content is required.	string
url *optional*	Optional url to the underlying content, should include version information, and can include a git hash. Note that this URL should resolve to the raw unwrapped content that would otherwise be available in content. One of url or content is required. **Example** : ""	string

3.5. ImageData

Describes one container image.

Name	Description	Schema
checksum *optional*	A production (immutable) tool version is required to have a hashcode. Not required otherwise, but might be useful to detect changes. This exposes the hashcode for specific image versions to verify that the container version pulled is actually the version that was indexed by the registry. **Example** : [{ "checksum" : "77af4d6b9913e693e8d0b4b294fa62ade6054e6b2f1ffb617ac955dd63fb0182", "type" : "sha256" }]	< Checksum > array
image_name *optional*	Used in conjunction with a registry_url if provided to locate images. **Example** : ""	string
image_type *optional*		ImageType
registry_host *optional*	A docker registry or a URL to a Singularity registry. Used along with image_name to locate a specific image. **Example** : ""	string

Name	Description	Schema
size *optional*	Size of the container in bytes.	integer
updated *optional*	Last time the container was updated.	string

3.6. ImageType

Indicates what kind of container is this image is.

Type : enum (Docker, Singularity, Conda)

3.7. Tool

A tool (or described tool) is defined as a tuple of a descriptor file (which potentially consists of multiple files), a set of container images, and a set of instructions for creating those images.

Name	Description	Schema
aliases *optional*	Support for this parameter is optional for tool registries that support aliases. A list of strings that can be used to identify this tool which could be straight up URLs. This can be used to expose alternative ids (such as GUIDs) for a tool for registries. Can be used to match tools across registries.	< string > array
checker_url *optional*	Optional url to the checker tool that will exit successfully if this tool produced the expected result given test data.	string
description *optional*	The description of the tool.	string
has_checker *optional*	Whether this tool has a checker tool associated with it.	boolean
id *required*	A unique identifier of the tool, scoped to this registry. **Example** : `"123456"`	string
meta_version *optional*	The version of this tool in the registry. Iterates when fields like the description, author, etc. are updated.	string
name *optional*	The name of the tool.	string

Name	Description	Schema
organization *required*	The organization that published the image.	string
toolclass *required*		ToolClass
url *required*	The URL for this tool in this registry. **Example** : `"http://agora.broadinstitute.org/tools/123456"`	string
versions *required*	A list of versions for this tool.	< ToolVersion > array

3.8. ToolClass

Describes a class (type) of tool allowing us to categorize workflows, tasks, and maybe even other entities (such as services) separately.

Name	Description	Schema
description *optional*	A longer explanation of what this class is and what it can accomplish.	string
id *optional*	The unique identifier for the class.	string
name *optional*	A short friendly name for the class.	string

3.9. ToolFile

Name	Description	Schema
file_type *optional*		enum (TEST_FILE, PRIMARY_DESCRIPTOR, SECONDARY_DESCRIPTOR, CONTAINERFILE, OTHER)
path *optional*	Relative path of the file. A descriptor's path can be used with the GA4GH .../{type}/descriptor/{relative_path} endpoint.	string

3.10. ToolVersion

A tool version describes a particular iteration of a tool as described by a reference to a specific image and/or documents.

Name	Description	Schema
author *optional*	Contact information for the author of this version of the tool in the registry. (More complex authorship information is handled by the descriptor).	< string > array
containerfile *optional*	Reports if this tool has a containerfile available. (For Docker-based tools, this would indicate the presence of a Dockerfile)	boolean
descriptor_type *optional*	The type (or types) of descriptors available.	< DescriptorType > array
id *required*	An identifier of the version of this tool for this particular tool registry. **Example** : `"v1"`	string
images *optional*	All known docker images (and versions/hashes) used by this tool. If the tool has to evaluate any of the docker images strings at runtime, those ones cannot be reported here.	< ImageData > array
included_apps *optional*	An array of IDs for the applications that are stored inside this tool. **Example** : [`"https://bio.tools/tool/mytum.de/SNAP2/1"`, `"https://bio.tools/bioexcel_seqqc"`]	< string > array
is_production *optional*	This version of a tool is guaranteed to not change over time (for example, a tool built from a tag in git as opposed to a branch). A production quality tool is required to have a checksum	boolean
meta_version *optional*	The version of this tool version in the registry. Iterates when fields like the description, author, etc. are updated.	string
name *optional*	The name of the version.	string
signed *optional*	Reports whether this version of the tool has been signed.	boolean

Name	Description	Schema
url *required*	The URL for this tool version in this registry. **Example** : "http://agora.broadinstitute.org/tools/123456/versions/1 "	string
verified *optional*	Reports whether this tool has been verified by a specific organization or individual.	boolean
verified_sourc e *optional*	Source of metadata that can support a verified tool, such as an email or URL.	< string > array