Using Glance's Metadata Definitions Catalog Public APIs

« (glanceclient.html) » (signature.html) (https://bugs.launchpad.net/glance/+filebug?field.title=Using%20Glance%E2%80%99s%20Metadata%20Definitions%20Catalog%20Public%20APIs%20in%20glance&field.comment=%0A%0A%0AThis bug tracker is for errors with the documentation, use the following as a template and remove or add fields as you see fit. Convert [ ] into [x] to check boxes:%0A%0A- [ ] This doc is inaccurate in this way: _____%0A- [ ] This is a doc addition request.%0A- [ ] I have a fix to the document that I can paste below including example: input and output. %0A%0AIf you have a troubleshooting or support issue, use the following resources:%0A%0A - Ask OpenStack: http://ask.openstack.org%0A - The mailing list: http://lists.openstack.org%0A - IRC: 'openstack' channel on Freenode%0A%0A----------%0ARelease:%2016.0.1.dev1%20on%20'Thu%20Mar%201%2007:26:57%202018,%20commit%20968f4ae'%0ASHA:%20968f4ae9ce244d9372cb3e8f45acea9d557f317d%0ASour https://docs.openstack.org/glance/queens/user/glancemetadefcatalogapi.html&field.tags=)

UPDATED: 'THU MAR 1 07:26:57 2018, COMMIT 968F4AE'

A common API hosted by the Glance service for vendors, admins, services, and users to meaningfully define available key / value pair and tag metadata. The intent is to enable better metadata collaboration across artifacts, services, and projects for OpenStack users.

This is about the definition of the available metadata that can be used on different types of resources (images, artifacts, volumes, flavors, aggregates, etc). A definition includes the properties type, its key, it's description, and it's constraints. This catalog will not store the values for specific instance properties.

For example, a definition of a virtual CPU topology property for number of cores will include the key to use, a description, and value constraints like requiring it to be an integer. So, a user, potentially through Horizon, would be able to search this catalog to list the available properties they can add to a flavor or image. They will see the virtual CPU topology property in the list and know that it must be an integer. In the Horizon example, when the user adds the property, its key and value will be stored in the service that owns that resource (Nova for flavors and in Glance for images).

Diagram: https://wiki.openstack.org/w/images/b/bb/Glance-Metadata-API.png (https://wiki.openstack.org/w/images/b/bb/Glance-Metadata-API.png)

Glance Metadata Definitions Catalog implementation started with API version v2.

Authentication¶

Glance depends on Keystone and the OpenStack Identity API to handle authentication of clients. You must obtain an authentication token from Keystone send it along with all API requests to Glance through the **`X-Auth-Token`** header. Glance will communicate back to Keystone to verify the token validity and obtain your identity credentials.

See Authentication With Keystone (../admin/authentication.html#authentication) for more information on integrating with Keystone.

Using v2.X¶

For the purpose of examples, assume there is a Glance API server running at the URL **`http://glance.openstack.example.org`** on the default port 80.

List Available Namespaces¶

We want to see a list of available namespaces that the authenticated user has access to. This includes namespaces owned by the user, namespaces shared with the user and public namespaces.

We issue a **`GET`** request to **`http://glance.openstack.example.org/v2/metadefs/namespaces`** to retrieve this list of available namespaces. The data is returned as a JSON-encoded mapping in the following format:

```
{
  "namespaces": [
      {
          "namespace": "MyNamespace",
          "display_name": "My User Friendly Namespace",
          "description": "My description",
          "visibility": "public",
          "protected": true,
          "owner": "The Test Owner",
          "self": "/v2/metadefs/namespaces/MyNamespace",
          "schema": "/v2/schemas/metadefs/namespace",
          "created_at": "2014-08-28T17:13:06Z",
          "updated_at": "2014-08-28T17:13:06Z",
          "resource_type_associations": [
              {
                  "name": "OS::Nova::Aggregate",
                  "created_at": "2014-08-28T17:13:06Z",
                  "updated_at": "2014-08-28T17:13:06Z"
              },
              {
                  "name": "OS::Nova::Flavor",
                  "prefix": "aggregate_instance_extra_specs:",
                  "created_at": "2014-08-28T17:13:06Z",
                  "updated_at": "2014-08-28T17:13:06Z"
              }
          ]
      }
  ],
  "first": "/v2/metadefs/namespaces?sort_key=created_at&sort_dir=asc",
  "schema": "/v2/schemas/metadefs/namespaces"
}
```


Note

Listing namespaces will only show the summary of each namespace including counts and resource type associations. Detailed response including all its objects definitions, property definitions etc. will only be available on each individual GET namespace request.

Filtering Namespaces Lists¶

`GET /v2/metadefs/namespaces` requests take query parameters that serve to filter the returned list of namespaces. The following list details these query parameters.

- **`resource_types=RESOURCE_TYPES`**

  Filters namespaces having a **`resource_types`** within the list of comma separated **`RESOURCE_TYPES`**.

GET resource also accepts additional query parameters:

- **`sort_key=KEY`**

  Results will be ordered by the specified sort attribute **`KEY`**. Accepted values include **`namespace`**, **`created_at`** (default) and **`updated_at`**.

- **`sort_dir=DIR`**

  Results will be sorted in the direction **`DIR`**. Accepted values are **`asc`** for ascending or **`desc`** (default) for descending.

- **`marker=NAMESPACE`**

  A namespace identifier marker may be specified. When present only namespaces which occur after the identifier **`NAMESPACE`** will be listed, i.e. the namespaces which have a *sort_key* later than that of the marker **`NAMESPACE`** in the *sort_dir* direction.

- **`limit=LIMIT`**

  When present the maximum number of results returned will not exceed **`LIMIT`**.

Note

If the specified **`LIMIT`** exceeds the operator defined limit (api_limit_max) then the number of results returned may be less than **`LIMIT`**.

- **`visibility=PUBLIC`**

  An admin user may use the *visibility* parameter to control which results are returned (PRIVATE or PUBLIC).

Retrieve Namespace¶

We want to see a more detailed information about a namespace that the authenticated user has access to. The detail includes the properties, objects, and resource type associations.

We issue a **`GET`** request to **`http://glance.openstack.example.org/v2/metadefs/namespaces/{namespace}`** to retrieve the namespace details. The data is returned as a JSON-encoded mapping in the following format:

```
{
  "namespace": "MyNamespace",
  "display_name": "My User Friendly Namespace",
  "description": "My description",
  "visibility": "public",
  "protected": true,
  "owner": "The Test Owner",
  "schema": "/v2/schemas/metadefs/namespace",
  "resource_type_associations": [
      {
          "name": "OS::Glance::Image",
          "prefix": "hw_",
          "created_at": "2014-08-28T17:13:06Z",
          "updated_at": "2014-08-28T17:13:06Z"
      },
      {
          "name": "OS::Cinder::Volume",
          "prefix": "hw_",
          "properties_target": "image",
          "created_at": "2014-08-28T17:13:06Z",
          "updated_at": "2014-08-28T17:13:06Z"
      },
      {
          "name": "OS::Nova::Flavor",
          "prefix": "filter1:",
          "created_at": "2014-08-28T17:13:06Z",
          "updated_at": "2014-08-28T17:13:06Z"
      }
  ],
  "properties": {
      "nsprop1": {
          "title": "My namespace property1",
          "description": "More info here",
          "type": "boolean",
          "default": true
      },
      "nsprop2": {
          "title": "My namespace property2",
          "description": "More info here",
          "type": "string",
          "default": "value1"
      }
  },
  "objects": [
      {
          "name": "object1",
          "description": "my-description",
          "self": "/v2/metadefs/namespaces/MyNamespace/objects/object1",
          "schema": "/v2/schemas/metadefs/object",
          "created_at": "2014-08-28T17:13:06Z",
          "updated_at": "2014-08-28T17:13:06Z",
          "required": [],
          "properties": {
              "prop1": {
                  "title": "My object1 property1",
                  "description": "More info here",
                  "type": "array",
                  "items": {
                      "type": "string"
                  }
              }
          }
      },
      {
          "name": "object2",
          "description": "my-description",
          "self": "/v2/metadefs/namespaces/MyNamespace/objects/object2",
          "schema": "/v2/schemas/metadefs/object",
          "created_at": "2014-08-28T17:13:06Z",
          "updated_at": "2014-08-28T17:13:06Z",
          "properties": {
              "prop1": {
                  "title": "My object2 property1",
                  "description": "More info here",
                  "type": "integer",
                  "default": 20
              }
          }
      }
  ]
}
```


Retrieve available Resource Types¶

We want to see the list of all resource types that are available in Glance

We issue a **GET** request to **http://glance.openstack.example.org/v2/metadefs/resource_types** to retrieve all resource types.

The data is returned as a JSON-encoded mapping in the following format:

```
{
  "resource_types": [
      {
          "created_at": "2014-08-28T17:13:04Z",
          "name": "OS::Glance::Image",
          "updated_at": "2014-08-28T17:13:04Z"
      },
      {
          "created_at": "2014-08-28T17:13:04Z",
          "name": "OS::Cinder::Volume",
          "updated_at": "2014-08-28T17:13:04Z"
      },
      {
          "created_at": "2014-08-28T17:13:04Z",
          "name": "OS::Nova::Flavor",
          "updated_at": "2014-08-28T17:13:04Z"
      },
      {
          "created_at": "2014-08-28T17:13:04Z",
          "name": "OS::Nova::Aggregate",
          "updated_at": "2014-08-28T17:13:04Z"
      },
      {
          "created_at": "2014-08-28T17:13:04Z",
          "name": "OS::Nova::Server",
          "updated_at": "2014-08-28T17:13:04Z"
      }
  ]
}
```

Retrieve Resource Types associated with a Namespace¶

We want to see the list of resource types that are associated for a specific namespace

We issue a **GET** request to **http://glance.openstack.example.org/v2/metadefs/namespaces/{namespace}/resource_types** to retrieve resource types.

The data is returned as a JSON-encoded mapping in the following format:

```
{
  "resource_type_associations" : [
      {
         "name" : "OS::Glance::Image",
         "prefix" : "hw_",
         "created_at": "2014-08-28T17:13:04Z",
         "updated_at": "2014-08-28T17:13:04Z"
      },
      {
         "name" :"OS::Cinder::Volume",
         "prefix" : "hw_",
         "properties_target" : "image",
         "created_at": "2014-08-28T17:13:04Z",
         "updated_at": "2014-08-28T17:13:04Z"
      },
      {
         "name" : "OS::Nova::Flavor",
         "prefix" : "hw:",
         "created_at": "2014-08-28T17:13:04Z",
         "updated_at": "2014-08-28T17:13:04Z"
      }
  ]
}
```

Add Namespace¶

We want to create a new namespace that can contain the properties, objects, etc.

We issue a **POST** request to add an namespace to Glance:

```
POST http://glance.openstack.example.org/v2/metadefs/namespaces/
```

The input data is an JSON-encoded mapping in the following format:

```
{
  "namespace": "MyNamespace",
  "display_name": "My User Friendly Namespace",
  "description": "My description",
  "visibility": "public",
  "protected": true
}
```

> **Note**
>
> Optionally properties, objects and resource type associations could be added in the same input. See GET Namespace output above(input will be similar).

Update Namespace¶

We want to update an existing namespace

We issue a **PUT** request to update an namespace to Glance:

```
PUT http://glance.openstack.example.org/v2/metadefs/namespaces/{namespace}
```

The input data is similar to Add Namespace

Delete Namespace¶

We want to delete an existing namespace including all its objects, properties etc.

We issue a **DELETE** request to delete an namespace to Glance:

```
DELETE http://glance.openstack.example.org/v2/metadefs/namespaces/{namespace}
```

Associate Resource Type with Namespace¶

We want to associate a resource type with an existing namespace

We issue a **POST** request to associate resource type to Glance:

```
POST http://glance.openstack.example.org/v2/metadefs/namespaces/{namespace}/resource_types
```

The input data is an JSON-encoded mapping in the following format:

```
{
        "name" :"OS::Cinder::Volume",
        "prefix" : "hw_",
        "properties_target" : "image",
        "created_at": "2014-08-28T17:13:04Z",
        "updated_at": "2014-08-28T17:13:04Z"
}
```

Remove Resource Type associated with a Namespace¶

We want to de-associate namespace from a resource type

We issue a **DELETE** request to de-associate namespace resource type to Glance:

```
DELETE http://glance.openstack.example.org/v2//metadefs/namespaces/{namespace}/resource_types/{resource_type}
```

List Objects in Namespace¶

We want to see the list of meta definition objects in a specific namespace

We issue a **GET** request to **`http://glance.openstack.example.org/v2/metadefs/namespaces/{namespace}/objects`** to retrieve objects.

The data is returned as a JSON-encoded mapping in the following format:

```
{
      "objects": [
      {
          "name": "object1",
          "description": "my-description",
          "self": "/v2/metadefs/namespaces/MyNamespace/objects/object1",
          "schema": "/v2/schemas/metadefs/object",
          "created_at": "2014-08-28T17:13:06Z",
          "updated_at": "2014-08-28T17:13:06Z",
          "required": [],
          "properties": {
              "prop1": {
                  "title": "My object1 property1",
                  "description": "More info here",
                  "type": "array",
                  "items": {
                      "type": "string"
                  }
              }
          }
      },
      {
          "name": "object2",
          "description": "my-description",
          "self": "/v2/metadefs/namespaces/MyNamespace/objects/object2",
          "schema": "/v2/schemas/metadefs/object",
          "created_at": "2014-08-28T17:13:06Z",
          "updated_at": "2014-08-28T17:13:06Z",
          "properties": {
              "prop1": {
                  "title": "My object2 property1",
                  "description": "More info here",
                  "type": "integer",
                  "default": 20
              }
          }
      }
  ],
  "schema": "/v2/schemas/metadefs/objects"
}
```

Add object in a specific namespace¶

We want to create a new object which can group the properties

We issue a **POST** request to add object to a namespace in Glance:

```
POST http://glance.openstack.example.org/v2/metadefs/namespaces/{namespace}/objects
```

The input data is an JSON-encoded mapping in the following format:

```
{
  "name": "StorageQOS",
  "description": "Our available storage QOS.",
  "required": [
      "minIOPS"
  ],
  "properties": {
      "minIOPS": {
          "type": "integer",
          "description": "The minimum IOPs required",
          "default": 100,
          "minimum": 100,
          "maximum": 30000369
      },
      "burstIOPS": {
          "type": "integer",
          "description": "The expected burst IOPs",
          "default": 1000,
          "minimum": 100,
          "maximum": 30000377
      }
  }
}
```

Update Object in a specific namespace¶

We want to update an existing object

We issue a **PUT** request to update an object to Glance:

```
PUT http://glance.openstack.example.org/v2/metadefs/namespaces/{namespace}/objects/{object_name}
```

The input data is similar to Add Object

Delete Object in a specific namespace¶

We want to delete an existing object.

We issue a **DELETE** request to delete object in a namespace to Glance:

```
DELETE http://glance.openstack.example.org/v2/metadefs/namespaces/{namespace}/objects/{object_name}
```

Add property definition in a specific namespace¶

We want to create a new property definition in a namespace

We issue a **POST** request to add property definition to a namespace in Glance:

```
POST http://glance.openstack.example.org/v2/metadefs/namespaces/{namespace}/properties
```

The input data is an JSON-encoded mapping in the following format:

```
{
  "name": "hypervisor_type",
  "title" : "Hypervisor",
  "type": "array",
  "description": "The type of hypervisor required",
  "items": {
      "type": "string",
      "enum": [
          "hyperv",
          "qemu",
          "kvm"
      ]
  }
}
```

Update property definition in a specific namespace¶

We want to update an existing object

We issue a **PUT** request to update an property definition in a namespace to Glance:

```
PUT http://glance.openstack.example.org/v2/metadefs/namespaces/{namespace}/properties/{property_name}
```

The input data is similar to Add property definition

Delete property definition in a specific namespace¶

We want to delete an existing object.

We issue a **DELETE** request to delete property definition in a namespace to Glance:

```
DELETE http://glance.openstack.example.org/v2/metadefs/namespaces/{namespace}/properties/{property_name}
```

API Message Localization¶

Glance supports HTTP message localization. For example, an HTTP client can receive API messages in Chinese even if the locale language of the server is English.

How to use it¶

To receive localized API messages, the HTTP client needs to specify the **Accept-Language** header to indicate the language to use to translate the message. For more info about Accept-Language, please refer http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html (http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html)

A typical curl API request will be like below:

```
curl -i -X GET -H 'Accept-Language: zh' -H 'Content-Type: application/json'
http://glance.openstack.example.org/v2/metadefs/namespaces/{namespace}
```

Then the response will be like the following:

```
HTTP/1.1 404 Not Found
Content-Length: 234
Content-Type: text/html; charset=UTF-8
X-Openstack-Request-Id: req-54d403a0-064e-4544-8faf-4aeef086f45a
Date: Sat, 22 Feb 2014 06:26:26 GMT

<html>
<head>
<title>404 Not Found</title>
</head>
<body>
<h1>404 Not Found</h1>
&#25214;&#19981;&#21040;&#20219;&#20309;&#20855;&#26377;&#26631;&#35782; aaa &#30340;&#26144;&#20687;<br /><br />
</body>
</html>
```

Note

Be sure there is the language package under /usr/share/locale-langpack/ on the target Glance server.

« (glanceclient.html) » (signature.html) (https://bugs.launchpad.net/glance/+filebug?field.title=Using%20Glance%E2%80%99s%20Metadata%20Definitions%20Catalog%20Public%20APIs%20in%20glance&field.comment=%0A%0A%0AThis bug tracker is for errors with the documentation, use the following as a template and remove or add fields as you see fit. Convert [ ] into [x] to check boxes:%0A%0A- [ ] This doc is inaccurate in this way: _____%0A- [ ] This is a doc addition request.%0A- [ ] I have a fix to the document that I can paste below including example: input and output. %0A%0AIf you have a troubleshooting or support issue, use the following resources:%0A%0A - Ask OpenStack: http://ask.openstack.org%0A - The mailing list: http://lists.openstack.org%0A - IRC: 'openstack' channel on Freenode%0A%0A----------%0ARelease:%2016.0.1.dev1%20on%20'Thu%20Mar%201%2007:26:57%202018,%20commit%20968f4ae'%0ASHA:%20968f4ae9ce244d9372cb3e8f45acea9d557f317d%0ASo https://docs.openstack.org/glance/queens/user/glancemetadefcatalogapi.html&field.tags=)

UPDATED: 'THU MAR 1 07:26:57 2018, COMMIT 968F4AE'

FOUND AN ERROR? REPORT A BUG (HTTPS://BUGS.LAUNCHPAD.NET/GLANCE/+FILEBUG?FIELD.TITLE=USING%20GLANCE%E2%80%99S%20METADATA%20DEFINITIONS%20CATALOG%20PUBLIC%20APIS%20IN%20GLANCE&FIELD.COMMENT=%0A%0A%0ATHIS BUG TRACKER IS FOR ERRORS WITH THE DOCUMENTATION, USE THE FOLLOWING AS A TEMPLATE AND REMOVE OR ADD FIELDS AS YOU SEE FIT. CONVERT [ ] INTO [X] TO CHECK BOXES:%0A%0A- [ ] THIS DOC IS INACCURATE IN THIS WAY: _____%0A- [ ] THIS IS A DOC ADDITION REQUEST.%0A- [ ] I HAVE A FIX TO THE DOCUMENT THAT I CAN PASTE BELOW INCLUDING EXAMPLE: INPUT AND OUTPUT. %0A%0AIF YOU HAVE A TROUBLESHOOTING OR SUPPORT ISSUE, USE THE FOLLOWING RESOURCES:%0A%0A - ASK OPENSTACK: HTTP://ASK.OPENSTACK.ORG%0A - THE MAILING LIST: HTTP://LISTS.OPENSTACK.ORG%0A - IRC: 'OPENSTACK' CHANNEL ON FREENODE%0A%0A----------%0ARELEASE:%2016.0.1.DEV1%20ON%20'THU%20MAR%201%2007:26:57%202018,%20COMMIT%20968F4AE'%0ASHA:%20968F4AE9CE244D9372CB3E8F45ACEA9D557F317D%0ASOURCE:%20HTTPS://HTTPS://DOCS.OPENSTACK.ORG/GLANCE/QUEENS/USER/GLANCEMETADEFCATALOGAPI.HTML&FIELD.TAGS=)

QUESTIONS? (HTTP://ASK.OPENSTACK.ORG)

OpenStack Documentation

glance 16.0.1.dev1

(../index.html)

User guide (index.html)
- Image Identifiers (identifiers.html)
- Image Statuses (statuses.html)
- Task Statuses (statuses.html#task-statuses)
- Disk and Container Formats (formats.html)
- Common Image Properties (common-image-properties.html)
- Metadata Definition Concepts (metadefs-concepts.html)
- Using Glance's Image Public APIs (glanceapi.html)
- Using Glance's Client Tools (glanceclient.html)
- Using Glance's Metadata Definitions Catalog Public APIs
- Image Signature Verification (signature.html)

Administration guide (../admin/index.html)
Installation (../install/index.html)
Glance Configuration Options (../configuration/index.html)
Command Line Interface (../cli/index.html)
Glance Contribution Guidelines (../contributor/index.html)

Glossary (../glossary.html)

Page Contents