

# HTTP REST API for OpenStack Object Storage Metadata Search (OSMS)

---

## Revision History

Version	Date	Author	Changes
0.1	19-Mar-2013	Lincoln Thomas	Initial draft
0.2	23-Jul-2013	Lincoln Thomas	Cumulative changes from comments and work by OSMS team so far (mostly Tomer Meshorer and Kimberly Keeton). Better alignment with OpenStack Object Storage API.
0.3	5-Aug-2013	Lincoln Thomas	Updates from OSMS team review meetings and further investigation.
0.4	21-Aug-2013	Lincoln Thomas	Updates from OSMS team review meetings and further investigation. More explanations and details.
0.5	1-Sep-2013	Lincoln Thomas	Updates during design and implementation phases. Some simplifications, more explanations, and example queries added.
0.6	2-Oct-2013	Lincoln Thomas	Minor wording changes.
0.7	7-Oct-2013	Lincoln Thomas	Minor clarifications per comments from Perry Merritt.
0.8	5-Nov-2013	Lincoln Thomas	Further updates per comments from Perry and team review.

## Introduction

This document specifies a set of REST API additions to OpenStack's Object Storage API to support a new metadata search feature, abbreviated OSMS, for Object Storage Metadata Search.

The OSMS API extends the set of features described in the base OpenStack Object Storage API, which can be found at <http://docs.openstack.org/api/openstack-object-storage/1.0>.

The OSMS feature maintains metadata for Object Storage accounts, containers, and objects. OSMS provides an HTTP REST API with a rich set of metadata search parameters for users and applications to search the stored metadata.

We use the term "item" to refer to an account, container, or object, when the distinction is unnecessary.

The HTTP REST request defines the base URI to search, the search criteria, the list of metadata fields to return for the items matching the criteria, and how to filter and format the resulting output.

## System and custom metadata

Two types of metadata are supported for searches, and both can be referenced in the same search:

- System metadata applies to all items (accounts, containers, and objects). Each item stored includes a fixed set of attributes comprising its system metadata. System metadata attributes cannot be deleted by the user through the API.
- Custom metadata applies only to items where the user assigns them using the "X-[Account|Container|Object]-Meta-<key>: <value>" HTTP header. Custom metadata names are user-defined, with value strings also defined by the user. Custom metadata can be added, replaced, or deleted by the user as specified in the OpenStack API. Custom metadata attributes are distinguished from system metadata attributes by the [account|container|object]\_meta\_prefix.

## Metadata attributes available

The following table describes the system and custom metadata attributes available for searches using the OSMS API.

For descriptions of the Types column, see the section "Data types."

### Live and deleted items

With the exception of the `*delete_time` attributes, all of the system metadata attributes listed in this table are valid for live (not-yet-deleted) items. For deleted items, only the following attributes are valid: `*uri`, `*name`, `*delete_time`, and `*last_activity_time`.

By default, only results for live items will be returned. To include deleted items in the results, include the appropriate `*delete_time` system attribute(s), either as part of the attribute list to be returned, or as a query criterion. For example, including `delete_time` or `account_delete_time` will cause account results to include live and deleted items that meet the query criteria to be returned.

System attribute	Type	Description	Example value
<code>account_uri</code>	string	An account in URI format, which is a slash character followed by the <code>account_name</code> , with no trailing slash.	<code>/ABSmith</code>
<code>account_name</code>	string	The name of an account, as defined by the authentication mechanism. Matches the account name provided in HTTP requests.	<code>ABSmith</code>
<code>account_tenant_id</code>	string	The name of the OpenStack Keystone	<code>AlbertBSmith</code>

		tenant that maps to this account. If Keystone is not used as the authentication mechanism, this attribute will be empty.	
account_first_use_time	date	<p>The time at which the account was first referenced by a base Object Storage API request.</p> <p>An account is never explicitly added by an API request. It is added when an API request first references the account. For example, creating the first container in an account causes a new account record to be added to the object store. Other examples include listing containers or adding custom metadata to an account, if the account has never been referenced before.</p>	2013-06-09T19:02:22.359070125Z
account_last_modified_time	date	The most recent time at which the list of containers changed. It does not include system nor custom metadata changes on the account.	2013-06-09T19:02:22.359070125Z
account_last_changed_time	date	The most recent time at which any metadata about the account changed, or the list of containers changed.	2013-06-09T19:02:22.359070125Z
account_delete_time	date	The time at which the account was removed from the object store.	2013-06-09T19:02:22.359070125Z
account_last_activity_time	date	<p>For live accounts, equivalent to account_last_changed_time.</p> <p>For deleted accounts, equivalent to</p>	2013-06-09T19:02:22.359070125Z

		<p><b>account_delete_time.</b></p> <p>The attribute provides a single source for the time of the last action upon an account, regardless of whether it is live or deleted. It can be used to query for all changes made to this account (including deletions) since a specified point in time.</p> <p>Deleted accounts are only queried or returned according to the rules in the section "Live and deleted items".</p>	
account_container_count	numeric	The number of containers in this account.	12
account_object_count	numeric	The number of objects in all containers in this account.	23439
account_bytes_used	numeric	<p>The number of bytes used by all objects in all containers in this account.</p> <p>Containers themselves do not contribute to the bytes used, only their objects.</p>	948654143
account_meta_<name>	string	<p>The value of the account custom metadata attribute &lt;name&gt;.</p> <p>Matches the "X-Account-Meta-&lt;name&gt;:&lt;value&gt;" header in HTTP requests.</p>	<p><b>Example attribute name:</b> account_meta_UserGroup</p> <p><b>Example value:</b> Administrators</p>
container_uri	string	A container in URI format, which is an account_uri, then a slash, then the container_name, with no trailing slash.	/ABSmith/container1
container_name	string	The name of a container. Matches the container name provided in HTTP requests.	container1
container_account_name	string	The name of the account	ABSmith

e		that includes this container.	
container_create_time	date	The time at which the container was created.	2013-06-09T19:02:22.359070125Z
container_last_modified_time	date	The most recent time at which the list of objects changed. It does not include system nor custom metadata changes on the container.	2013-06-09T19:02:22.359070125Z
container_last_change_d_time	date	The most recent time at which any metadata about the container changed, or the list of objects changed.	2013-06-09T19:02:22.359070125Z
container_delete_time	date	The time at which the container was deleted.	2013-06-09T19:02:22.359070125Z
container_last_activity_time	date	For live containers, equivalent to container_last_changed_time.  For deleted containers, equivalent to container_delete_time.  Refer to the account_last_activity_time attribute and the section "Live and deleted items" for details.	2013-06-09T19:02:22.359070125Z
container_read_permissions	string	The value of the X-Container-Read header, obtained from the HTTP request when the container is created or modified.  Empty if no such header is ever supplied.	.r:*,.r:-.thief.com,bobs_account,sues_account:sue,.rlistings
container_write_permissions	string	The value of the X-Container-Write header, obtained from the HTTP request when the container is created or modified.	bobs_account,sues_account:sue

		Empty if no such header is ever supplied.	
<code>container_sync_to</code>	string	<p>The container URL to mirror this container's contents to, for container sync. Includes the full HTTP URL with cluster name and API version followed by the container-level URI.</p> <p>Matches the X-Container-Sync-To header in HTTP requests when container sync is configured.</p> <p>Empty if no such header is ever supplied.</p>	<code>http://cluster2/v1/ABSmith/container2</code>
<code>container_sync_key</code>	string	<p>A secret key string defined during container sync to allow this container's contents to be mirrored to another container that accepts this key.</p> <p>Matches the X-Container-Sync-Key header in HTTP requests when container sync is configured.</p> <p>Empty if no such header is ever supplied.</p>	<code>mySecretKey</code>
<code>container_versions_location</code>	string	<p>The container URI containing previous versions of objects in this container, if object versioning is configured.</p> <p>Matches the X-Versions-Location header in HTTP requests when container sync is configured.</p> <p>Empty if no such header is ever supplied.</p>	<code>/ABSmith/container1oldVersions</code>

container_object_count	numeric	The number of objects in this container.	768
container_bytes_used	numeric	The number of bytes used by all objects in this container.  Containers themselves do not contribute to the bytes used, only their objects.	32459978
container_meta_<name>	string	The value of the container custom metadata attribute <name>.  Matches the "X-Container-Meta-<name>:<value>" header in HTTP requests.	<b>Example attribute name:</b> container_meta_department  <b>Example value:</b> sales
object_uri	string	An object in URI format, which is a container_uri, then a slash character, then the object_name, with no trailing slash.	/ABSmith/container1/objectdir1/subdir1/photo.jpg  In this example, the object_name is objectdir1/subdir1/photo.jpg  See the object_name description below.
object_name	string	The name of an object, which must be a string that is unique within a container. Matches the object name provided in HTTP requests.  Note that the object_name may contain a "pseudo path" of directory names and a file name, as in this example, or it may contain any arbitrary unique string such as "myObject".  When pseudo-paths are used, the path is only meaningful to the client. No such directory	objectdir1/subdir1/photo.jpg

		hierarachy exists in the object store, and the path is unrelated to where the object is stored on the object store's internal file system.	
object_account_name	string	The name of the account that includes this object.	ABSmith
object_container_name	string	The name of the container that includes this object.	container1
object_location	string	<p>The URL of the location of the object on the storage system. The URL may be accessible from the client or may be accessible only from the storage system.</p> <p>This attribute is intended to be useful to administrators that manage an object store. If an implementation chooses not to expose this attribute, for security or other reasons, it can omit this attribute from the attr_list of the services request. If omitted, any client use of this attribute will return in the HTTP error code 405 (Method Not Allowed) in the response. See the section "Services request" for details.</p>	<pre>file:///fs1/ObjectStoreRoot/node103104105106/objects/576/0c9/240dcb5fac9bb402f06b43a3b1ae50c9/1376071675.13258.data</pre> <p>In this example, the URL is a file location accessible only to users logged in to one of the server nodes of the storage system.</p>
object_uri_create_time	date	<p>The time at which the first object with this URI was created.</p> <p>If an object with this URI does not already exist when the object is uploaded with a PUT, then this attribute is assigned the same value as object_last_modified_time.</p>	2013-06-09T19:02:22.359070125Z

		If an object with this URI already exists when the object is uploaded, then this attribute is not changed.	
object_last_modified_time	date	The most recent time at which the object contents changed. It does not include system nor custom metadata changes on the object.	2013-06-09T19:02:22.359070125Z
object_last_changed_time	date	The most recent time at which any metadata about the object changed.	2013-06-09T19:02:22.359070125Z
object_delete_time	date	The time at which the object was deleted.	2013-06-09T19:02:22.359070125Z
object_last_activity_time	date	For live objects, equivalent to object_last_changed_time.  For deleted objects, equivalent to object_delete_time.  Refer to the account_last_activity_time attribute and the section "Live and deleted items" for details.	2013-06-09T19:02:22.359070125Z
object_etag_hash	string	The MD5 checksum of the object contents (or of the concatenated string of the segments' MD5 checksums, for a manifest object), as generated by the Object Storage code in response to the base API's Create/Update Object Request. The response to that request includes the ETag HTTP header containing this checksum value.  In the base API's List	d9f5eb4bba4e2f2f046e54611bc8196b

		Objects response, this checksum is shown as the "hash" attribute in the JSON or XML output.	
object_content_type	string	The MIME type of the object, as supplied by the client in the Content-Type header in the HTTP request.	text/plain; charset=UTF-8
object_content_length	numeric	The size of the object in bytes.  If the object is a manifest, the size will be the sum of the sizes of all of the object's segments.	43572
object_content_encoding	string	The compression method, if supplied by the client in the Content-Encoding header in the HTTP request.	gzip
object_content_disposition	string	Defines the browser's download behavior, if supplied by the client in the Content-Disposition header in the HTTP request.	attachment; filename=platmap.tif
object_content_language	string	Defines the natural language(s) of the intended audience for the object, if supplied in the Content-Language header in the HTTP request.	mi, en
object_cache_control	string	Defines caching mechanisms to be obeyed by all caching mechanisms along the request / response chain, if supplied in the Cache-Control header in the HTTP request.	max-age=2844, no-transform
object_delete_at	date	The time at which the object should be deleted.  Converted from the UNIX epoch time in the X-Delete-At header in the HTTP request.	2013-06-09T19:02:22.359070125Z

		<p>Converted from the <code>X-Delete-After</code> header of the HTTP request, which defines the integer number of seconds from now when the object should be deleted.</p> <p>For either conversion, the resulting date is the same time format as the other times (ISO 8601:2004 format).</p> <p>If both <code>X-Delete-At</code> and <code>X-Delete-After</code> appear in the same HTTP request, the one that appears later in the request is the one that will be stored.</p>	
<code>object_manifest_type</code>	numeric	<p>If 0 (zero), this object is not a manifest.</p> <p>If 1, this object is a dynamic large object (DLO).</p> <p>If 2, this object is a static large object (SLO).</p>	2
<code>object_manifest</code>	string	<p>For DLOs uploaded in segments, the value is the <code>X-Object-Manifest</code> header supplied by the client in the HTTP request. The string is a container name followed by a slash character, then the common object names prefix for all object segments.</p> <p>For SLOs, the value is the entire JSON manifest generated by Swift based on the body of the uploaded SLO manifest object.</p>	<p>For a DLO:</p> <pre>container1/objectdir1/subdir/photosegments</pre> <p>For an SLO:</p> <pre>{ "path": "/container1/object1",   "etag": "d9f5eb4bba4e2f2f046e54611bc8196b",   "size_bytes": 1048576}, ...]</pre>
<code>object_access_control</code>	string	Cross-Object Resource	See the <a href="http://w3.org">w3.org</a> page.

<pre> _allow_origin object_access_control _allow_credentials object_access_control _expose_headers object_access_control _max_age object_access_control _allow_methods object_access_control _allow_headers object_origin object_access_control _request_method object_access_control _request_headers </pre>		<p>Sharing (CORS) attributes, as described in <a href="http://www.w3.org/TR/access-control/#access-control-allow-origin-response-header">http://www.w3.org/TR/access-control/#access-control-allow-origin-response-header</a></p>	
<pre>object_meta_&lt;name&gt;</pre>	string	<p>The value of the object custom metadata attribute &lt;name&gt;.</p> <p>Matches the "X-Object-Meta-&lt;name&gt;:&lt;value&gt;" header in HTTP requests.</p>	<p><b>Example attribute name:</b> object_meta_department</p> <p><b>Example value:</b> sales</p>

## Superset metadata attributes available

The OSMS API provides the following special attribute names that can be requested in the `&attributes` parameter list, to return a superset of certain system and/or custom attributes. These attributes cannot be used as query criteria or as sort parameters.

The same rules about live and deleted files that apply to system attributes apply to these attributes as well.

Superset attribute	Description
all_attrs	Returns all system and custom metadata attributes for the items matching the query. Equivalent to <code>all system attrs, all meta attrs</code> .
all_system_attrs	Returns all system metadata attributes except <code>*delete_time</code> , for the items matching the query. It does not include any custom metadata attributes. Equivalent to <code>all_account_system_attrs, all_container_system_attrs, all_object_system_attrs</code> . The system attributes returned will be appropriate to the item. If the item is an account, only account system metadata will be returned. Containers and objects likewise return only the system metadata appropriate to those items.

<code>all_meta_attrs</code>	Returns all custom (user-defined) metadata attributes of the form "[account container object]_meta_<name>" for the items matching the query. It does not include any system metadata attributes.
<code>all_account_attrs</code>	Returns all system and custom metadata attributes for the accounts matching the query (but not for containers nor objects). Equivalent to <code>all_account_system_attrs,all_account_meta_attrs</code> .
<code>all_account_system_attrs</code>	Returns all system metadata attributes except <code>*delete_time</code> , for the accounts matching the query. It does not include any custom metadata attributes.
<code>all_account_meta_attrs</code>	Returns all custom (user-defined) metadata attributes of the form "account_meta_<name>" for the accounts matching the query. It does not include any system metadata attributes.
<code>all_container_attrs</code>	Same as <code>all_account_attrs</code> but for containers.
<code>all_container_system_attrs</code>	Same as <code>all_account_system_attrs</code> but for containers.
<code>all_container_meta_attrs</code>	Same as <code>all_account_meta_attrs</code> but for containers.
<code>all_object_attrs</code>	Same as <code>all_account_attrs</code> but for objects.
<code>all_object_system_attrs</code>	Same as <code>all_account_system_attrs</code> but for objects.
<code>all_object_meta_attrs</code>	Same as <code>all_account_meta_attrs</code> but for objects.

## HTTP syntax

The HTTP request format for obtaining the list of services for this implementation is the following:

```
GET /services HTTP/1.1
```

The HTTP request for performing metadata searches has the following syntax, as one request but shown on multiple lines for readability:

```
GET /<Object Storage API version>
[/<account>[/<container>[/<object>]]]
?<OSMS API version>
[&attributes=<attr1>[,<attr2>][, ...]]
[&query=[ (<query
expr1>[%20[AND|OR]%20<query expr2>] ) ] [%20[AND|OR]%20...]]
[&sorted[=<attr1>[,<attr2>, ...]]]
[&limit=<max records>]
[&all_results]
[&marker='<marker string>']
[&end_marker='<end marker string>']
[&offset=<start record>]
[&prefix='<prefix string>']
[&path='<object directory path>']
[&delimiter='<delimiter character>']
[&format=[json|xml]] HTTP/1.1
```

The equivalent `curl` command formats for the two request types are the following, each as one command line:

```
curl -g "http[s]://<IP address|hostname>[:<port>]
/services

curl -g "http[s]://<IP address|hostname>[:<port>]
/<Object Storage API version>
[/<account>/<container>/<object>]]]
?<OSMS API version>
[&attributes=<attr1>[,<attr2>][, ...]]
[&query=[(<query expr1>[ [AND|OR] <query expr2>][)][ [AND|OR]...]]
[&sorted=[<attr1>[,<attr2>, ...]]]
[&limit=<max records>]
[&all_results]
[&marker='<marker string>']
[&end_marker='<end marker string>']
[&offset=<start record>]
[&prefix='<prefix string>']
[&path='<object directory path>']
[&delimiter='<delimiter character>']
[&format=[json|xml]]"
```

### Syntax notes

- Only the HTTP GET verb is used for metadata searches.
- Optional parameters are shown in square brackets [ and ]. Everything enclosed in the brackets can be omitted from the request. Do not include the square brackets in the request. For example, the `query` parameter is optional and is not required to be in the HTTP request.
- Parameters are shown in angle brackets < and >. Replace the parameter with the actual value, without the angle brackets.
- Other characters shown in the syntax (such as =, ?, &, and /) must also be entered as-is in the request, and sometimes must be URL-encoded.

### Quoting

- Numeric values for parameters must not be quoted.
- Date values for parameters must be enclosed in single quotes, e.g.:  
'2013-06-09Z'
- Reserved strings, including API versions, attribute names, and format types, must not be quoted, e.g.:  
&attributes=objectLastModifiedTime,objectContentType
- User-defined string values for parameters must be enclosed in single quotes, e.g.:  
prefix='employees/'
- Any single quotes that are part of a quoted string value must be escaped with a second single quote. For example:  
'Dave''s book'
- In the `curl` syntax shown above, the `-g` option and the double quotes around the entire URL ensure that the URL contents will be parsed correctly by `curl`. Any double quotes that are part

of a quoted string value in a double quoted curl command must be escaped with a backslash:  
the `\\"right\"` way

## URL encoding

HTTP request strings are URL-decoded by the API code. API clients must encode special characters, such as greater-than character (>), by replacing them with their hexadecimal equivalent values as shown by the examples in this section.

The API's URL decoder interprets certain special characters properly without being URL encoded. Before the question mark character (?) in any HTTP request URL, the following characters are safe and do not need to be URL encoded:

`/ : - _ . ~ @ #`

After the question mark character (?), the following characters are safe and do not need to be URL encoded:

`= & #`

All other characters must be URL encoded as their hexadecimal value as described in the ISO-8859-1 (ISO-Latin) standard. For example, the plus character (+) must be encoded as `%2B`, and the greater than character (>) must be encoded as `%3E`.

Spaces can be encoded as either `%20` or as the plus character (+), such as `"my%20file.txt"` or `"my+file.txt"` for the file `"my file.txt"`. The plus character (+) is converted to a space when the URL is decoded by the API code. To include a plus character (+) in the URL, encode it as `%2B`, such as `"A%2B"` instead of `"A+"`.

## Data types

The type of each attribute will be one of:

string	A series of characters meeting the given <charset> in the Content-Type HTTP header. If <charset> or Content-Type is not present in the HTTP header, UTF-8 will be assumed. No null terminators or null characters will be present. An empty string is represented by no characters.
numeric	Zero or a positive integer value less than 2 <sup>64</sup> (18,446,744,073,709,551,616). In HTTP requests and responses, the value is represented as a string as described above.
date	See the "Date formats" section below.

## Date formats

All date/time values accepted by the API in HTTP requests, must be in ISO 8601:2004 format. See [http://www.iso.org/iso/catalogue\\_detail?csnumber=40874](http://www.iso.org/iso/catalogue_detail?csnumber=40874) or [http://en.wikipedia.org/wiki/ISO\\_8601](http://en.wikipedia.org/wiki/ISO_8601) for details on the format.

Examples of values accepted in HTTP requests:

```
'2013-06-09'
```

(9-Jun-2013, at time 00:00:00Z, where Z = Zulu a.k.a. GMT a.k.a. UTC time zone)

```
'2013-06-09T09:02:26Z'
```

(9-Jun-2013 at 9:02:22am in Zulu/ GMT/UTC time zone)

```
'20130609T090226Z'
```

(The same date/time without separators)

```
'2013-06-09T02:02:26-0700'
```

(9-Jun-2013 at 2:02:22am in time zone -0700 = 7 hours behind UTC, which could be e.g. Pacific Daylight Time, or Mountain Standard Time. Equivalent to T09:02:22Z)

Example value not accepted in HTTP requests:

```
'2013-06-09T09:02:26'
```

(No timezone indicator, will return an error)

```
'Mon, 17 Oct 2011 14:31:11 GMT'
```

(RFC 5322 format, not ISO 8601 format, will return an error)

```
'1346895723.552374000'
```

(UNIX epoch format, not ISO 8601 format, will return an error)

All date/time values returned in HTTP responses also will be in ISO 8601:2004 format, and will always be in the UTC time zone.

The granularity of the metadata time values stored and returned in HTTP responses may vary depending on how the value is stored and obtained. The granularity of a given time value may be in nanoseconds, microseconds, or in seconds only. The fractional seconds shown in the response, however, will always be in nanoseconds.

Example values returned include:

```
"2013-06-09T19:02:22.000000000Z"
```

(seconds granularity)

```
"2013-06-09T19:02:22.817493000Z"
```

(microseconds granularity)

```
"2013-06-09T19:02:22.359070125Z"
```

(nanoseconds granularity)

## URIs and metadata search scope

A URI uniquely defines an account, container, or object. In an HTTP request, the URI to the left of the '?' character defines the scope of the metadata search.

The URI can specify any of these combinations to the left of the '?':

1. An account, container, and object, known as an object-level URI. Searches will operate only upon that object.
2. An account and container only, known as a container-level URI. Searches will be limited to the container and the objects within it.
3. An account only, known as an account-level URI. Searches will be limited to the account, and all containers and objects within it.
4. None of these three parameters. Searches will operate on all accounts, containers, and objects in the object store.

The search behavior depends on the parameters specified in the request. See the table below for details on each of these parameters.

## HTTP request parameters

Parameter	Description
IP_address	The dotted decimal IP address of the Object Storage proxy server that will handle this HTTP request and return an HTTP response, e.g. 10.11.12.13
hostname	The host name of the Object Storage proxy server that will handle this HTTP request and return an HTTP response, e.g. proxy1 or proxy1.mycompany.com.
port	The listener IP port of the Object Storage proxy server. If the proxy is listening on the default HTTP port of 80, or the default HTTPS port 443, this parameter is unnecessary.
services	Requests the list of services provided by the server's implementation of the base Object Storage API and the Metadata Search API. See the section "Services request" below this table for further information.
Object Storage API version	Indicates the version of the OpenStack Object Storage API that this OSMS API is extending. This element is specified in the Object Storage API. As of this writing, the only version that exists is "v1".
account	<p>The name of the account to search.</p> <p>If present, the search scope will be limited to that account. If absent, all accounts will be searched.</p> <p>The following access rules apply. See the section "Authorized searchers" below this table for further information on the authorized searcher feature.</p> <p>If absent:</p> <p style="padding-left: 40px;">If the searching user is a member of authorized searchers, then access will be provided to all metadata for all accounts and all of their containers and objects.</p> <p style="padding-left: 40px;">If the searching user is not an authorized searcher, then access to each account's metadata will depend on the OpenStack authentication mechanism approving access to the account in the URI for the given <code>X-Auth-Token</code>. For each account that the user can access, she has access to the metadata of that account and all containers and objects in that account. For accounts inaccessible to the user, she can only access object metadata in those containers that provide <code>.rlistings</code> access in their</p>

	<p><code>X-Container-Read</code> permission, and no access to any container or account metadata for those accounts.</p> <p>If present:</p> <p>If the searching user is a member of authorized searchers, then access will be provided to all metadata for that account and all of its containers and objects.</p> <p>If the searching user is not an authorized searcher, then access to that account's metadata will depend on the OpenStack authentication mechanism approving access to the account in the URI for the given <code>X-Auth-Token</code>. If the user has access to the account, she has access to the metadata of the account and all containers and objects in that account. If the user does not have access to the account, she can only access object metadata in those containers that have <code>.rlistings</code> permission in their <code>X-Container-Read</code> ACLs, and no access to any container or account metadata.</p>
container	<p>The name of the container to search.</p> <p>If present, the search scope will be limited to that container. If absent, see the description of the <code>&lt;account&gt;</code> parameter.</p> <p>The following access rules apply:</p> <p>If the searching user is a member of authorized searchers, then access will be provided to all metadata for that container, its parent account, and all objects in that container. Account metadata does not include its container list.</p> <p>If the searching user is not an authorized searcher, then metadata access will depend on the OpenStack authentication mechanism approving access to the <code>&lt;container&gt;</code> in the URI for the given <code>X-Auth-Token</code>. If the user has access to the container's account, he will be able to access the metadata for the <code>&lt;account&gt;</code>, for the <code>&lt;container&gt;</code>, and for all objects in that container. If the user does not have access to the account, but the container has <code>.rlistings</code> permission in the <code>X-Container-Read</code> ACL that includes the searching user, he will be able to access the metadata for all objects in that container, but not the container's nor the account's metadata.</p>
object	<p>The name of the object to search.</p> <p>If present, the search scope will be limited to that container. If absent, see the description of the <code>&lt;container&gt;</code> parameter.</p> <p>The following access rules apply:</p> <p>If the searching user is a member of authorized searchers, then access will be provided to the metadata for the <code>&lt;account&gt;</code>, for the <code>&lt;container&gt;</code>, and for the <code>&lt;object&gt;</code>. Account metadata does not include its container list, and container</p>

	<p>metadata does not include its object list.</p> <p>If the searching user is not an authorized searcher, then metadata access to this object will depend on the authentication mechanism approving access to the &lt;container&gt; in the URI for the given <code>X-Auth-Token</code>. If the user has access to the container's account, she will be able to access the metadata for the &lt;account&gt;, for the &lt;container&gt;, and for the &lt;object&gt;. If the user does not have access to the account but has <code>.rlistings</code> permission in the <code>X-Container-Read</code> ACL for the container, she will be able to access the metadata for the &lt;object&gt;, but not the container's nor the account's metadata.</p>
OSMS API version	<p>Indicates the version of the OSMS API. As of this writing, always "v1". Separating the OSMS API version from the OpenStack Object Storage API version allows the OSMS API to change independently of the Object Storage API that it extends. Retaining the version element of the Object Storage API maintains compatibility with that API's URI specification.</p>
<pre>attributes= &lt;attr1&gt;[,&lt;attr2&gt;][,...]</pre>	<p>If present, a comma-separated list of system, custom, and/or superset attribute names to be returned in the response for each item matching the query criteria.</p> <p>The URI of every item returned will always be shown in the item's attribute list, whether or not its attribute is listed, even if &lt;attributes&gt; is absent. The URI is the unique identification of every item returned.</p> <p>Besides defining which attributes to return for the items in the result set, &lt;attributes&gt; can also filter the result set:</p> <p style="padding-left: 40px;">If any <code>account_*</code> attributes are listed, or any superset attributes that include account attributes are listed, then any account items in the result set will be returned as entries in the response. Otherwise, no account items will be returned.</p> <p style="padding-left: 40px;">Similar filtering is done for <code>container_*</code> and <code>object_*</code> attributes.</p> <p style="padding-left: 40px;">For example, if the URI <code>/acc1/ctr2</code> is given, and only object &lt;attr&gt;s are listed, then only the entries for objects from the result set will be returned. If only account and container &lt;attr&gt;s are listed, then only two entries at most will be returned: one for <code>acc1</code> (if it exists in the result set), and one for <code>ctr2</code> (if it exists in the result set).</p> <p>If absent, then every item in the result set will be returned, but only the <code>*_uri</code> attribute will be shown.</p> <p>The entries from the result set that will be returned in the response may be further limited by other request parameters such as &lt;limit&gt;.</p>
query expr	<p>Defines the result set of items, even if it is absent.</p> <p>The result set begins with the scope defined by the URI in the request, before applying a query. See the section "URIs and metadata search scope". The items in the scope's pre-query result set include the URI item, its parent items, and all</p>

	<p>items in lower-level URIs. For example, the pre-query result set for a container-level URI includes all objects in the container, the container itself, and its parent account.</p> <p>If any &lt;query expr&gt;s are present:</p> <p style="padding-left: 40px;">The result set is further limited to the items meeting all &lt;query expr&gt; expressions.</p> <p>A &lt;query expr&gt; compares an attribute against a given value. The format of a query expression is: &lt;query attr&gt;&lt;operator&gt;&lt;query value&gt; See the next table rows below for details on these parameters.</p> <p>Query expressions can be combined using one or more AND and OR Boolean operators (case-insensitive), as well as parentheses to override default order of operations rules. There must be one space character (%20) before and after each AND or OR. By default, all AND expressions will be processed (left to right) before all OR expressions. Then the OR expressions will be processed (left to right). See the "Example queries" section for examples.</p> <p>Other Boolean operators such as NOT, NAND, and XOR are not supported. The NOT operator is not needed because each expression can be converted to its opposite to achieve the same effect. Other operators can be achieved using combinations of AND, OR, and parentheses.</p> <p>An implementation can choose to support a simpler subset of Boolean operator support. See the Services section for details.</p> <p>If &lt;attributes&gt; is absent, then the result set will be filtered further, based on the &lt;query attr&gt;s in the &lt;query expr&gt;s:</p> <p style="padding-left: 40px;">If any account_* attributes are specified in any &lt;query attr&gt;, then any accounts shown in any URIs in the result set will be included as items in the filtered result set. Otherwise, no accounts will be included in the filtered result set.</p> <p style="padding-left: 40px;">If any container_* attributes are specified in any &lt;query attr&gt;, then any containers shown in any URIs in the result set will be included as items in the filtered result set. Otherwise, no containers will be included in the filtered result set.</p> <p style="padding-left: 40px;">If any object_* attributes are specified in any &lt;query attr&gt;, then all objects in the result set will be included as items in the filtered result set. Otherwise, no objects will be included in the filtered result set.</p> <p>If no &lt;query expr&gt;s are present, the result set is defined by the URI scope. For example, if the URI /acc1/ctr2 is given, then the result set includes an entry for</p>
--	--

	<p>acc1, an entry for ctr2, and an entry for each object in ctr2.</p> <p>The result set may be further filtered by other request parameters such as &lt;attributes&gt; or &lt;limit&gt;.</p>
<pre>query attr (element of &lt;query expr&gt;)</pre>	<p>A system and/or custom metadata attribute to be compared against the &lt;query value&gt; as a query criterion.</p>
<pre>operator (element of &lt;query expr&gt;)</pre>	<p>The query operation to perform against the &lt;query attr&gt; and &lt;query value&gt;, one of:</p> <ul style="list-style-type: none"> <li>= (equals exactly)</li> <li>!= (does not equal)</li> <li>&lt; (less than)</li> <li>&lt;= (less than or equal to)</li> <li>&gt; (greater than)</li> <li>&gt;= (greater than or equal to)</li> </ul> <p>Only for custom attributes and string-valued system attributes (for example, object_container_name):</p> <ul style="list-style-type: none"> <li>~ (regular expression match)</li> <li>!~ (does not match regular expression)</li> </ul> <p>The arguments to the regular expression operators (~ and !~) are POSIX regular expressions, as described in POSIX 1003.1-2008 at <a href="http://pubs.opengroup.org/onlinepubs/9699919799/">http://pubs.opengroup.org/onlinepubs/9699919799/</a>, section 9, Regular Expressions.</p>
<pre>query value (element of &lt;query expr&gt;)</pre>	<p>The value to compare against the &lt;query attr&gt; using the &lt;operator&gt;. The value is either a numeric decimal value without quotes, or a string enclosed in single quotes. See the "Quoting" section above for details.</p>
<pre>sorted[= &lt;attr1&gt;[, &lt;attr2&gt;, ...]]</pre>	<p>If this parameter is present with no value, the list of items returned is sorted lexicographically by URI. Accounts, containers, and items will be arranged hierarchically, such as:</p> <pre>/a1 /a1/c1 /a1/c1/o1 /a1/c1/o2 /a1/c2 /a2 {etc.}</pre> <p>The use of &lt;sorted&gt; on large result sets might negatively affect the performance of the query due to the sort processing required.</p> <p>If this parameter is present with one or more named system or custom metadata attributes, the list of items returned is sorted by that list of attributes, first by attr1, then by attr2, and so forth. String attributes are sorted lexicographically. Numeric attributes are sorted in increasing numeric value. Date attributes are sorted by increasing timestamp.</p> <p>If &lt;sorted&gt; is absent, items will be returned by item levels. First, all accounts</p>

	<p>meeting the query criteria are returned. Next, all containers meeting the criteria are returned. Next, all objects meeting the criteria are returned. Each level's results are returned in an undefined order, but the order is repeatable if none of the items in the result set are added or deleted between two identical queries.</p>
<pre>limit= &lt;max records&gt;</pre>	<p>If this parameter is present, at most &lt;max records&gt; items will be returned in the response. The value is a numeric data type, see the "Data types" section.</p> <p>The result set is sorted first (or created in the default order), then the limit and other pagination attributes are applied.</p> <p>If absent, at most 10,000 items will be returned, by default. If other results are desired, use one or more of the parameters: &lt;all_results&gt;, &lt;marker&gt;, &lt;end_marker&gt;, &lt;offset&gt;.</p>
<pre>all_results</pre>	<p>If this parameter is present, then all results will be returned, overriding the default limit of 10,000 items. If &lt;limit&gt; is also present, then the &lt;all_results&gt; parameter is ignored.</p>
<pre>marker= '&lt;marker string&gt;'</pre>	<p>A pagination attribute. For all pagination attributes, note that a separate query will be issued for each HTTP request.</p> <p>If this parameter is present, then &lt;marker string&gt; is in the URI format: ' /&lt;account&gt;[/&lt;container&gt;[/&lt;object&gt;]] '</p> <p>Items that appear after the given URI in the result set will be returned in the response (not including the URI). The behavior is independent of the &lt;sorted&gt; parameter. The result set may be sorted or not, but the order of the results is repeatable if none of the items in the result set are added or deleted between two identical queries.</p> <p>There is no state saved between queries. If objects are added or deleted during the time that the client is issuing queries to get each page, the results from the set of all queries may not represent the complete set at any particular point in time. If no objects are added nor deleted during this time, then the results will include all items that match the query during that time.</p> <p>If the marker does not match any URI in the result set, and &lt;sorted&gt; is not present, an error 404 Not Found is returned, because it's unclear where the marker would have appeared in the result set.</p> <p>If &lt;offset&gt; is present, then &lt;marker&gt; is ignored.</p>
<pre>end_marker= '&lt;end marker string&gt;'</pre>	<p>A pagination attribute. If this parameter is present, then only items that appear before the given &lt;end marker string&gt; in the result set will be returned in the response.</p> <p>If &lt;offset&gt; is present, then &lt;end_marker&gt; is ignored.</p>

	See the <marker> parameter for further details.
offset= <start record>	<p>A pagination attribute. If this parameter is present, then results will be returned starting with &lt;start record&gt;. The first (&lt;start record&gt;-1) items from the result set are not returned. For example, if the &lt;start record&gt; is 100, then the first item returned will be the 100<sup>th</sup> item, and the prior 99 items in the result set will not be returned.</p> <p>The result set may be sorted or not, but the order of the results is repeatable if none of the items in the result set are added or deleted between two queries of different offsets.</p>
prefix= '<prefix string>'	<p>An object name filter attribute. If present and &lt;delimiter&gt; is absent, then the query will return only the objects in the result set where the &lt;object_name&gt; begins with the given &lt;prefix string&gt;.</p> <p>If &lt;delimiter&gt; is absent, the same functionality can be accomplished by specifying: &amp;query=object_name~'&lt;prefix string&gt;.*' (without the angle brackets).</p> <p>If &lt;delimiter&gt; is also present, see the &lt;delimiter&gt; description for details.</p> <p>If &lt;path&gt; is present, &lt;prefix&gt; and &lt;delimiter&gt; are ignored.</p>
delimiter= '<delimiter character>'	<p>An object name filter attribute. The &lt;delimiter&gt; parameter is used to limit the results of a hierarchy of &lt;object_name&gt;s to a certain level of nesting, as described in the base Object Storage API's two "Pseudo-Hierarchical Folders/Directories" sections.</p> <p>However, whereas the base API may return a partial object name that represents multiple objects, as a single result item, metadata search returns metadata for actual objects. Therefore, the output using &lt;delimiter&gt; differs from the List Objects output in the base API, but its use case is the same: All entries at the given hierarchy level will be returned. But if an entry contains objects at lower levels, metadata from one object from the lower levels will be returned, instead of just the partial object name returned by the base API.</p> <p>If present, and &lt;prefix&gt; is not present, then the query will return one object from each set of &lt;object_name&gt;s. The objects of a set begin with the same string up to the &lt;delimiter character&gt;. If no delimiter exists in the &lt;object_name&gt;, that object will be returned.</p> <p>If present, and &lt;prefix&gt; is also present, then the query will return one object from each set of &lt;object_name&gt;s. The objects of a set begin with &lt;prefix string&gt; and continue with the same string after the prefix, up to the &lt;delimiter character&gt;. Every object with no delimiter after the &lt;prefix string&gt; in the &lt;object_name&gt; will be returned.</p> <p>If &lt;sorted&gt; is present, then the object from each set will be the first object of that</p>

	<p>set sorted lexicographically by URI. If <code>&lt;sorted&gt;</code> is not present, the object from each set is undetermined but repeatable if none of the items in the result set are added or deleted between two identical queries.</p> <p>If <code>&lt;path&gt;</code> is present, <code>&lt;prefix&gt;</code> and <code>&lt;delimiter&gt;</code> are ignored.</p> <p>The combination of <code>&lt;prefix&gt;</code> and <code>&lt;delimiter&gt;</code>, or the use of <code>&lt;path&gt;</code> without <code>&lt;prefix&gt;</code> or <code>&lt;delimiter&gt;</code>, allows queries of a specified depth of pseudo-hierarchical directories used within object names.</p> <p>For example, suppose the following unsorted list of objects exists in a container:</p> <pre>employees/Eve employees/hourly/Ted employees/hourly/Deb employees/former/ employees2/Bob contractors/Joe</pre> <p><b>A query containing:</b> <code>...&amp;sorted&amp;prefix='employees/'&amp;delimiter='/'...</code> and no additional <code>&lt;query expr&gt;</code> criteria will return three objects sorted lexicographically:</p> <pre>employees/Eve employees/former/ employees/hourly/Deb</pre> <p>(Note that a <code>&lt;prefix&gt;</code> of <code>employees</code> without the trailing <code>'/'</code> would also return <code>employees2/Bob</code>.)</p> <p>In the base API, the same number of items would be returned, but the item names would be:</p> <pre>employees/Eve employees/former/ employees/hourly/</pre> <p>Metadata search cannot return <code>employees/hourly/</code> because it does not represent an actual object. Instead, it returns the metadata of one object that begins with that prefix, in this case <code>employees/hourly/Deb</code>.</p> <p>The same effect can be achieved using <code>&lt;path&gt;</code> instead of <code>&lt;prefix&gt;</code> and <code>&lt;delimiter&gt;</code>:</p> <pre>...&amp;sorted&amp;path='employees'...</pre> <p>Note the absence of the trailing <code>'/'</code> in the <code>&lt;path&gt;</code>. This equivalent usage is only possible because the <code>&lt;prefix&gt;</code> ends in a <code>'/'</code> and the <code>&lt;delimiter&gt;</code> is <code>'/'</code>.</p> <p>If <code>&lt;sorted&gt;</code> is not present in the above example, then the query will return objects in an undefined order, and the query will include either</p>
--	--

	<p>'employees/hourly/Ted' or '/employees/hourly/Deb', since the internal order is undefined. Because the order is repeatable, however, either Ted or Deb will always be returned if none of the items in the result set are added or deleted between two identical queries.</p> <p>The &lt;prefix&gt; description describes which objects in which containers and accounts will be queried.</p>
<pre>path= '&lt;object directory path&gt;'</pre>	<p>An object name filter attribute. If present, a '/' will be appended to the given &lt;object directory path&gt; parameter value string, unless it already contains a trailing '/'. Then the query will be processed exactly as if the &lt;prefix&gt; parameter contained the same &lt;object directory path&gt; with a '/' character at the end, and a &lt;delimiter&gt; of '/' was present.</p> <p>If present, &lt;prefix&gt; and &lt;delimiter&gt; are ignored.</p>
<pre>format= [json xml]</pre>	<p>If present, must be one of the string literals <code>json</code> or <code>xml</code>. Results are returned in JSON or XML format. If absent, plain text is returned.</p>

## Authorized searchers

The concept of authorized searchers provides a way to maximize query speed by bypassing the need to check the searching user's access to each account, for a set of users defined by the storage administrator. These users must authenticate via the OpenStack authentication mechanism to be able to perform any query, just like all users. Once authenticated, however, the authorized searcher can access any metadata for all accounts, containers, and objects via the API.

For users that are not listed as authorized searchers, the engine will check the access to each account and container to be queried, and will only return results for accounts and containers to which the user has access. This access checking can significantly impact the performance of the query.

A storage administrator defines this list of users that are authorized searchers using a management console or using CLI commands. The administrator may distribute different user credentials to different users, for example to allow each user to manage their own password, or for audit tracking.

The authorized searchers feature is optional. If not used, access to metadata will be limited for all users, according to the user's access to accounts and containers.

## Output formats

The default output format for the query results is plain text, if no "format" parameter is supplied.

### Plain Text format

The plain text output format is human readable, with indent levels of 4 spaces per indent. The top level has no indent, and always represents the URI of an item matching the query criteria. The requested attributes for that item are listed under the URI, indented.

If <sorted> or <sorted=uri> is specified, then the results will be in lexicographic order by URI. Example:

```

/account1
  account_container_count:15
/account1/container1
  container_last_modified_time:2013-07-23T13:17:55.435654031Z
/account1/container1/objectdir1/subdir1/photo.jpg
  object_last_changed_time:2012-12-02T00:53:29.849922518Z
  object_content_length:194532
/account1/container2
  container_last_modified_time:2013-07-23T13:17:55.435654031Z
/account1/container2/anotherObject
  object_last_changed_time:2012-12-02T00:53:29.849922518Z
  object_content_length:194532

```

If `<sorted>` or `<sorted=uri>` is not specified, then the results will not be sorted, or will be sorted by the attribute defined in `<sorted=attr>`. Example of unsorted output:

```

/account1
  account_container_count:15
/account1/container1
  container_last_modified_time:2013-07-23T13:17:55.435654031Z
/account1/container2
  container_last_modified_time:2013-07-23T13:17:55.435654031Z
/account1/container1/objectdir1/subdir1/photo.jpg
  object_last_changed_time:2012-12-02T00:53:29.849922518Z
  object_content_length:194532
/account1/container2/anotherObject
  object_last_changed_time:2012-12-02T00:53:29.849922518Z
  object_content_length:194532

```

Each line of output is terminated by a single UNIX-style end-of-line character, UTF-8 value 10 (0x0a).

## JSON format

The JSON output format for query responses conforms to standard and well-formed JSON. The first level always represents the URI of an item matching the query criteria. The requested attributes for that item are listed as the second level after the URI.

See the description of sorted vs. unsorted output in the "Plain text" section above.

Example for sorted output:

```

[
  {
    "/account1" :
    {
      "account_container_count" : "15"
    }
  },
  {
    "/account1/container1" :

```

```

    {
      "container_last_modified_time" : "2013-07-
23T13:17:55.435654031Z"
    }
  },
  {
    "/account1/container1/objectdir1/subdir1/photo.jpg" :
    {
      "object_last_changed_time" : "2012-12-02T00:53:29.849922518Z",
      "object_content_length" : 194532
    }
  },
  {
    "/account1/container2" :
    {
      "container_last_modified_time" : "2013-07-
23T13:17:55.435654031Z"
    }
  },
  {
    "/account1/container2/anotherObject" :
    {
      "object_last_changed_time" : "2012-12-02T00:53:29.849922518Z",
      "object_content_length" : 194532
    }
  }
}
]

```

## XML format

The XML output format for query responses conforms to standard and well-formed XML. The output is flat, not hierarchical. Objects are not nested in containers, and containers are not nested in accounts. Each item is provided separately, which allows for unsorted and sorted outputs based on arbitrary sorting criteria.

See the description of sorted vs. unsorted output in the "Plain text" section above.

Example for sorted output:

```

<?xml version="1.0" encoding="UTF-8"?>

<account uri="/account1">
  <account_container_count>15</account_container_count>
</account>
<container uri="/account1/container1">
  <container_last_modified_time>2013-07-
23T13:17:55.435654031Z</container_last_modified_time>
</container>
<object uri="/account1/container1/objectdir1/subdir1/photo.jpg">
  <object_last_changed_time>2012-12-
02T00:53:29.849922518Z</object_last_changed_time>

```

```
<object_content_length>194532</object_content_length>
</object>
<container uri="/account1/container2">
  <container_last_modified_time>2013-07-
23T13:17:55.435654031Z</container_last_modified_time>
</container>
<object uri="/account1/container2/anotherObject">
  <object_last_changed_time>2012-12-
02T00:53:29.849922518Z</object_last_changed_time>
  <object_content_length>194532</object_content_length>
</object>
```

## Example searches

### Get all metadata for all accounts, containers, and objects

```
curl -g "http://99.226.50.92/v1?v1&attributes=all_attrs"
```

The search returns an entry for every account, every container, and every object. Each entry contains all system attributes, and any custom attributes. The output format is plain text (see the section "Plain Text format").

Note that unless the user issuing the search is an authorized searcher, the search must authenticate the user against each account, and each container in non-authorized accounts. The results will contain metadata only from authorized accounts, and containers from non-authorized accounts that provide read permission to the user in the container read ACL. Thus, searches by non-authorized searchers across multiple accounts may be significantly slower than searches by authorized searchers.

### Get selected metadata for an object, its container, and its account

```
curl -g
"http://99.226.50.92/v1/acc1/ctr2/obj3?v1&attributes=account_container_
count,account_object_count,account_meta_billing_method,all_container_
system_attrs,all_object_meta_attrs&format=json"
```

The search returns first an entry for account acc1 with the three named attributes (two system attributes `account_container_count` and `account_object_count`, and one custom attribute `account_meta_billing_method`). The next entry is for the container ctr2, returning all system attributes (`all_container_system_attrs`) but no custom attributes. If the object obj3 has any custom metadata attributes, the next entry is for obj3 with its custom metadata but no system attributes (`all_object_meta_attrs`). If obj3 has no custom attributes, then no results are returned for obj3.

The output format is JSON (see the section "JSON format").

### Get metadata for an account, and objects meeting a set of criteria

```
curl -g
"http://99.226.50.92/v1/acc1?v1&attributes=account_object_count,object
```

```
_last_changed_time&query=container_create_time>2013-08-01 AND  
object_manifest_type=1 and object_manifest~'segctr1/.*'"
```

The search returns first the object count for the acc1 account. Next, it returns the last changed time for all objects in the account where the object's container was created after 1-Aug-2013 at 00:00:00 UTC and the object is a manifest for a DLO object that has segments in the container segctr1.

This example also shows how the result set is limited by the attributes list. Although a container attribute is present in the query, no container entries exist in the final result set because the attribute list contains no container attributes. But because the URI of each item is always returned, the account and container for each object will be shown.

Note that the 'AND' operator is case-insensitive.

### Identify accounts and containers meeting a set of criteria

```
curl -g "http://99.226.50.92/v1?v1&query=account_object_count<100 AND  
container_meta_customer_name~'.*'"
```

The search returns only the URIs for all accounts and containers (but not objects), where the account's object count is less than 100, and containers have the custom attribute "customer\_name" regardless of its value.

This example also shows how the result set is limited by the item types in the query expressions, in the absence of a list of <attributes> to be returned. Since no object attributes exist in the query, no object items are returned.

### Get metadata for all results, sorted by URI

```
curl -g  
"http://99.226.50.92/v1/acc1/ctr2?v1&attributes=account_container_coun  
t,object_content_type&sorted&all_results"
```

The search first returns the account container count for acc1. Next, it returns the content type of each object in ctr2, sorted by the object URI. The `all_results` parameter overrides the default limit of 10,000 items in the response. All objects in ctr2 will be returned, regardless of the number of objects in it. The client accepting the HTTP response must be able to receive an arbitrarily large amount of data in this case.

### Get the 3rd page of 100 results, sorted by multiple attributes

```
curl -g  
"http://99.226.50.92/v1/acc1?v1&attributes=all_system_attrs&query=obje  
ct_content_length>1000000&sorted=container_object_count,object_uri_c  
reate_time&limit=100&offset=201"
```

The search returns results 201 through 300 of the sorted result set defined by the scope and query. The results in total are sorted first by container object count, then by the object URI's first creation time.

Note that because the superset attribute `all_system_attrs` in the attributes list applies to all item levels, an entry for `acc1` will be returned, as well as entries for all containers and objects satisfying the query.

For example, suppose two containers exist, with 300 and 200 objects respectively. First, the system attributes for the `acc1` account is returned. Next, the second container is returned, followed by its objects that are greater than 1,000,000 bytes (of the 200 that exist), in order of the object URIs' creation times. Next, the first container and all of its objects satisfying the query (of the 300 that exist) are returned, sorted the same way.

### Get a portion of the sorted results, between the given URIs

```
curl -g
"http://99.226.50.92/v1/acc1?v1&attributes=all_system_attrs&sorted&marker='acc1/ctr13/obj_x'&end_marker='acc1/ctr17/obj_y'"
```

The search returns the account, containers, and objects in the `acc1` account, between the URIs `'acc1/ctr13/obj_x'` and `'acc1/ctr17/obj_y'` (non-inclusive) in the sorted (by URI) list of results.

## Services request

The `<services>` parameter requests the API versions and list of services provided by the server's implementation of the given Object Storage API version and Metadata Search API version. Following are the elements of the response.

This request was created with the intention of generalizing the OSMS API to become a standard OpenStack metadata search API. That API would have a reference implementation provided by OpenStack, but the API could also be implemented by other vendors supplying a search provider with different levels of API support. A search provider may not choose to implement all of the API, and/or may add elements to the API. This request defines the elements provided with the implementation. It allows clients to access elements of the API appropriate to different search providers on different OpenStack instances.

Element	Description
<code>min_base_api_version</code>	The minimum version of the Object Storage API accepted by the search provider's implementation, e.g. "v1".
<code>max_base_api_version</code>	The maximum version of the Object Storage API accepted by the search provider's implementation, e.g. "v2". As of this writing, only the "v1" version of the API exists.
<code>search_provider</code>	The string identifier of the entity providing the metadata search service on the server, e.g. "HP".
<code>search_enabled</code>	<p>If "true" (case-sensitive), then the metadata search feature is supported by this implementation, and the following fields are meaningful.</p> <p>If not "true", the string should be interpreted as false. In this case, no metadata search features are available at all, and any search request that is not understood by the base API will return an HTTP error response 405, "Method Not Allowed". All fields after this one in</p>

	the Services request are meaningless and should not be used.
<code>min_search_api_version</code>	The minimum version of the metadata search API for the search provider's implementation, e.g. "v1". Note that e.g. "v2" for one provider may list a different set of service elements than "v2" from another provider.
<code>max_search_api_version</code>	The maximum version of the metadata search API for the search provider's implementation, e.g. "v2".  Note that e.g. "v2" for one provider may list a different set of service elements than "v2" from another provider.
<code>freshness_complete</code>	If "true" (case-sensitive), then each HTTP response to a metadata search request will contain the following HTTP header in the response, in ISO 8601:2004 format. For example:  <code>X-Freshness-Complete-Date:2013-06-09T18:56:52.479174135Z</code>  The X-Freshness-Complete-Date indicates the time at which all operations on the object store have been stored in the metadata search database and are queryable.  If not "true", the string should be interpreted as false.
<code>freshness_partial</code>	If "true" (case-sensitive), then each HTTP response to a metadata search request will contain the following HTTP header in the response, in ISO 8601:2004 format. For example:  <code>X-Freshness-Partial-Date:2013-06-09T19:02:22.359070125Z</code>  The X-Freshness-Partial-Date indicates the latest time when any operations have been stored. Operations between the two timestamps may or may not be stored in the database and returned in query results. No operations after the partial date timestamp have been stored and will not be returned in query results.  If not "true", the string should be interpreted as false.
<code>complex_boolean_expr</code>	If "true" (case-sensitive), then ANDs, ORs, and parentheses (nested expressions) are supported as query operators by this implementation. This is the full PI support for Boolean expressions defined by the API.  If not "true", the string should be interpreted as false. In this case, only ANDs are supported, which makes all queries simpler to implement. Multiple ANDs in one expression must be supported.
<code>attr_list</code>	The list of system, and superset attributes that the server can return in an attribute list and can query on. If an attribute is not listed, then

	<p>the implementation does not support it and will return an error if referenced in an HTTP request.</p> <p>If a listed attribute is not one of the set defined on this API specification, further details should be provided in the search provider's documentation.</p> <p>Each attribute contains the following sub-elements.</p>
attr_name	The string name of the attribute.
data_type	The type of the attribute. See the section "Data types" for details.
sortable	<p>If "true" (case-sensitive), the attribute can be requested as the &lt;sort&gt; attribute value.</p> <p>If not "true", the string should be interpreted as false.</p>

Responses follow the following JSON format example. No other formats such as XML or plain text are supported.

```
[
  {
    "min_base_api_version" : "v1"
  },
  {
    "max_base_api_version" : "v1"
  },
  {
    "search_provider" : "HP"
  },
  {
    "search_enabled" : "true"
  },
  {
    "min_search_api_version" : "v2.1"
  },
  {
    "max_search_api_version" : "v2.1"
  },
  {
    "freshness_complete" : "true"
  },
  {
    "freshness_partial" : ""
  },
  {
    "complex_boolean_expr" : "false"
  },
  {
    "attributes" :
    {
      {
```

```
        "attr_name" : "account_uri"
      },
      {
        "data_type" : "string"
      },
      {
        "sortable" : "true"
      }
    },
    ... more attributes ...
    {
      {
        "attr_name" : "size"
      },
      {
        "data_type" : "numeric"
      },
      {
        "sortable" : "false"
      }
    },
  }
]
```