CIFER API Framework and SCIM API -- Point by Point Comparison

As of Q3 2015, the CIFER Project has been transitioned to the TIER-Data Structures and APIs Working Group.

CIFER API Framework vis-a-vis SCIM API and SCIM Schema

CIFER Framework

4. Authentication and Authorization

The framework does not specify authentication or authentication methods. Some acceptable methods include:

1. Basic auth over https
2. Client certificate auth over https
3. Encrypted and signed content over any transport

JF: There are many equally valid auth methods

Issue #6: Capability for Server to Indicate Authentication Methods

SCIM API

2. Authentication and Authorization
The SCIM protocol does not define a scheme for authentication and authorization therefore implementers are free to choose mechanisms appropriate to their use cases. The choice of authentication mechanism will impact interoperability. It is RECOMMENDED that clients be implemented in such a way that new authentication schemes can be deployed. Implementers SHOULD support existing authentication/authorization schemes. In particular, OAuth2[RFC6750] is RECOMMENDED. Appropriate security considerations of the selected authentication and authorization schemes SHOULD be taken. Because this protocol uses HTTP response status codes as the primary means of reporting the result of a request, servers are advised to respond to unauthorized or unauthenticated requests using the 401 response code in accordance with section 10.4.2 of Section 10.4.2 [RFC2616].

All examples assume OAuth2 bearer token [RFC6750]; e.g.,

GET /Users/2819c223-7f76-453a-919d-413861904646 HTTP/1.1
Host: example.com
Authorization: Bearer h480djs93hd8

The context of the request (i.e. the user for whom data is being requested) MUST be inferred by service providers.

CIFER Framework

5.1.1. URI syntax

A major version of a particular CIFER API must be indicated in the URI path (for example, "/v2/"). A major version will differentiate incompatible representations, parameters or headers. A major version change is also indicated by a change in field type. Any minor version will be indicated in the response metadata not in the URI path (see response section below).

JF: SCIM language seems OK

Issue #4: Local Versioning in Request

SCIM API

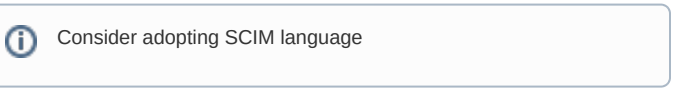
3.11. API Versioning

The Base URL MAY be appended with a version identifier as a separate segment in the URL path. At this time of this specification, the identifier is 'v2'. If specified, the version identifier MUST appear in the URL path immediately preceding the resource endpoint and conform to the following scheme: the character 'v' followed by the desired SCIM version number; e.g., a version 'v2' User request is specified as /v2/Users. When specified service providers MUST perform the operation using the desired version or reject the request. When omitted service providers SHOULD perform the operation using the most recent SCIM protocol version supported by the service provider.

CIFER Framework

5.1.2. Request/response grammar

1. Default representation language is Javascript Object Notation (JSON).
2. XML can be requested, but support is not required.



SCIM API

3. API

All requests to the service provider are made via Section 9 [RFC2616] on a URL derived from the Base URL. Responses are returned in the body of the HTTP response, formatted as JSON. Response and error codes SHOULD be transmitted via the HTTP status code of the response (if possible), and SHOULD also be specified in the body of the response.

3.6. Data Input/Output Formats

Servers MUST accept requests and respond with JSON structured responses using UTF-8 encoding [RFC3629], UTF-8 SHALL be the default encoding format.

Clients using other encodings MUST specify the format in which the data is submitted via Section 14.17 HTTP header content-type[RFC2616] and MAY specify the desired response data format via an HTTP Accept Header; e.g.," Accept: application/json" or via URI suffix; e.g.,

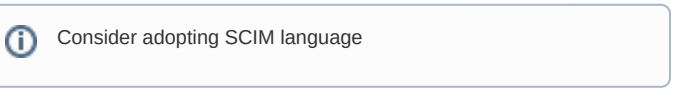
GET /Users/2819c223-7f76-453a-919d-413861904646.json
Host: example.com

Service providers MUST support the Accept Headers "Accept: application /json" for [RFC7159]. The format defaults to JSON if no format is specified.

CIFER Framework

5.1.3. Data and field formats

1. Character set is Unicode, encoded as UTF-8.



SCIM API

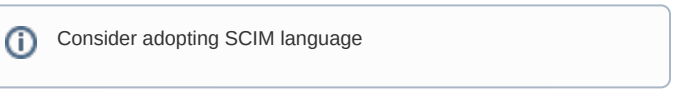
3.6. Data Input/Output Formats

> Servers MUST accept requests and respond with JSON structured responses using UTF-8 encoding [RFC3629], UTF-8 SHALL be the default encoding format.

CIFER Framework

5.1.3. Data and field formats

2. Timestamp representations will comply with ISO 8601. For example: 2012-10-04T03:10:14.123Z



SCIM Core Schema

3.1.5. DateTime

A DateTime value (e.g. 2008-01-23T04:56:22Z). The attribute value MUST be encoded as a valid xsd:dateTime as specified in Section 3.2.7 [XML-Schema].

Values represented in JSON MUST conform to the XML constraints above and are represented as a JSON String per Section 7 [RFC7159].
CIFER Framework

5.1.3. Data and field formats

3. Durations also comply with ISO 8601, For example: P1.81S
SCIM API

No mention of durations or time intervals
CIFER Framework

5.1.3. Data and field formats

4. Field names are camelCase (including acronyms), e.g. "selfUri"
5. Where there is discretion enum and other fixed field values should be camelCase.

> JF: We could go with SHOULDs in 4 and 5

SCIM Core Schema

3. SCIM Schema Structure

Attribute names SHOULD be camelCased.

No mention of rules over string-typed attribute values
CIFER Framework

5.1.3. Data and field formats

6. Field names must start with a letter. They can contain upper/lower letters, numbers and underscores.

> Consider adopting SCIM language

SCIM Core Schema

3. SCIM Schema Structure

Attribute names MUST conform to the following ABNF [RFC5234] rules:
ATTRNAME = ALPHA *(nameChar)
nameChar = "-" / "_" / DIGIT / ALPHA
CIFER Framework

5.1.3. Data and field formats

7. Field types must stay consistent within a major version.
SCIM

No guidance on what kinds of changes should lead to a new major version number
CIFER Framework

5.1.3. Data and field formats

8. Objects are identified by uri

- For example: netId:mchyzer, or userId:12345678

> JF: We could expand our commentary., but
>
> 1. It should be: "It SHOULD be a stable, non-reassignable identifier..
> 2. We don't need to except "bulkId"

SCIM Core Schema

5.1. Common Schema Attributes

id

Unique identifier for the SCIM resource as defined by the service provider. Each representation of the resource MUST include a nonempty "id" value. This identifier MUST be unique across the service provider's entire set of resources. It MUST be a stable, non-reassignable identifier that does not change when the same resource is returned in subsequent requests. The value of the "id" attribute is always issued by the service provider and MUST never be specified by the client. bulkId: is a reserved keyword and MUST NOT be used in the unique identifier. REQUIRED and has a mutability of "readOnly".
CIFER Framework

5.2.1 Resource references

CIFER APIs accommodate large responses by recognizing full and partial resources.

1. A full resource, where the response is the full resource requested, must be identified by a URI only, not by query string or fragment

2. A partial resource, like a response that is better served across pages, is identified by a URI that contains at least one additional query string parameter (see paging options in query string options section below)

NOTE: CIFER refers to pagination of single large resources, whereas SCIM discusses pagination only in reference to responses containing multiple resources.



Consider referencing OpenSearch protocol re pagination parameters and behavior

SCIM API

3.2.2.4 Pagination

Pagination parameters can be used together to "page through" large numbers of resources so as not to overwhelm the client or service provider. Pagination is not session based hence clients SHOULD never assume repeatable results. For example, a request for a list of 10 resources beginning with a startIndex of 1 may return different results when repeated as a resource in the original result could be deleted or new ones could be added in-between requests. Pagination parameters and general behavior are derived from the OpenSearch Protocol [OpenSearch].
CIFER Framework

5.2.2 Request Methods

1. The request method of POST [may be] overridden by a header (see header options below). NOTE: see section 5.2 3 below for details

2. A PUT request need not specify all attributes.

1. Replacing a resource will not affect attributes not specified in the request.
2. A necessary attribute not specified will return an error 400 (see response codes below)



JF: We have to consider missing attributes if we allow new attributes without re-versioning.

PATCH, if logically a set of PUT/DELETE, is OK

SCIM API

3. API

PUT Modifies a resource with a complete, client specified resource (replace).

PATCH Modifies a resource with a set of client specified changes (partial update).

NOTE: See also the substantial section 3.3 and its subsections on modifying resources

CIFER Framework

5.2.2. Request methods

3. Searches return resource URIs and default resource attributes.

- Each service will document the fields returns on searches.
- Searches may request "&extraFields=field3,field4,..&" to get those fields in addition to the default set.
- Searches may request "&omitFields=field3,field4,..&" to omit those fields from the default set.
- Support for these features is optional.

4. Searches may specify paging options by query string parameters

- pageLimit: Number of entries per page
- pageOffset: page number, first page is zero



Consider adopting SCIM language on queries (3.2.2 and 3.2.3)

SCIM API

3.2. Retrieving Resources

"User" and "Group" resources are retrieved via opaque, unique URLs or via Query. Service providers MAY choose to respond with a sub-set of resource attributes, though MUST minimally return the resource id and meta attributes.

3.2.1. Retrieving a known Resource

To retrieve a known resource, clients send GET requests to the resource endpoint; e.g., "/Users/{id}" or "/Groups/{id}".

If the resource exists the server responds with a status code of 200 and includes the result in the body of the response.

3.2.2. List/Query Resources

SCIM defines a standard set of operations that can be used to filter, sort, and paginate response results. The operations are specified by adding query parameters to the resource's endpoint. Service providers MAY support additional query parameters not specified here, and Providers SHOULD ignore any query parameters they don't recognize.

NOTE: See the rest of section 3.2.2 and its subsections for details on query endpoints, filters, sorting and pagination

3.2.3 Querying Resources Using HTTP POST

Clients MAY execute queries without passing parameters on the URL by using the HTTP POST verb combined with the '/.search' path extension. The inclusion of '/.search' on the end of a valid SCIM endpoint SHALL be used to indicate the HTTP POST verb is intended to be a query operation.

...

The following attributes are defined for search requests:

- attributes A multi-valued list of strings indicating the names of resource attributes to return in the response overriding the set of attributes that would be returned by default. Attribute names MUST be in standard attribute notation (Section 3.8) form. See additional retrieval query parameters (Section 3.7). OPTIONAL.
- excludedAttributes A multi-valued list of strings indicating the names of resource attributes to be removed from the default set of attributes to return. This parameter SHALL have no effect on attributes whose schema "returned" setting is "always" see Server Schema [I-D.ietf-scim-core-schema]. Attribute names MUST be in standard attribute notation (Section 3.8) form. See additional retrieval query parameters (Section 3.7). OPTIONAL.

CIFER Framework

5.2.3. Header Options

1. The request method of POST may be overridden by the header:

- X-HTTP-Method-Override: GET | PUT | DELETE

JF: A method override is sometimes necessary

SCIM API

Appendix C. Change log

Draft 05 - PH - Revisions based on the following tickets:

65 - Removed X-HTTP-Method-Override support

CIFER Framework

5.2.3. Header Options

2. XML representation may requested by header:

- Accept: text/xml

SCIM API

3. API

Responses are returned in the body of the HTTP response, formatted as JSON.

NOTE: SCIM has no provision for returning XML responses

CIFER Framework

5.2.3. Header Options

3. Use of ETag and If-Match:

- A request to replace a resource must specify an If-Match header
  - The value must match the resource's current ETag value, or
  - The value may be a 'match-any' wildcard, '*'
  - Acceptance of wildcard is service dependent.
  - A request to add a resource must not specify an If-Match header

Review in light of SCIM approach

SCIM API

3.12. Versioning Resources

The API supports resource versioning via standard HTTP ETags Section 2.3 [RFC7233]. Service providers MAY support weak ETags as the preferred mechanism for performing conditional retrievals and ensuring clients do not inadvertently overwrite each others changes, respectively. When supported SCIM ETags MUST be specified as an HTTP header and SHOULD be specified within the 'version' attribute contained in the resource's 'meta' attribute.

...

If the service providers supports versioning of resources the client MAY supply an If-Match Section 3.1 [RFC7233] header for PUT and PATCH operations to ensure that the requested operation succeeds only if the supplied ETag matches the latest service provider resource; e.g., If-Match: W /"e180ee84f0671b1"

CIFER Framework

5.2.3. Header Options

4. A client may request to "act as" another user via a header:

- X-CIFER-Act-As: user_uri

SCIM API

NOTE: SCIM does not have a concept of authorization of proxy actors such as that represented by the X-CIFER-Act-As header

CIFER Framework

5.2.4. Query String Options

1. Requests may specify field filters on returned resources.

- Query string may specify a list desired fields: "&fields=field1,field2,... &"
- A requests (sic) for an unknown field is not an error.
- Support for this feature is optional.



Review in light of SCIM approach

SCIM API

3.2.2. List/Query Resources

SCIM defines a standard set of operations that can be used to filter, sort, and paginate response results. The operations are specified by adding query parameters to the resource's endpoint. Service providers MAY support additional query parameters not specified here, and Providers SHOULD ignore any query parameters they don't recognize.

NOTE: See the rest of section 3.2.2 and its subsections for details on query endpoints, filters, sorting and pagination

3.2.3 Querying Resources Using HTTP POST

Clients MAY execute queries without passing parameters on the URL by using the HTTP POST verb combined with the '/.search' path extension. The inclusion of '/.search' on the end of a valid SCIM endpoint SHALL be used to indicate the HTTP POST verb is intended to be a query operation.

...

The following attributes are defined for search requests:

- attributes A multi-valued list of strings indicating the names of resource attributes to return in the response overriding the set of attributes that would be returned by default. Attribute names MUST be in standard attribute notation (Section 3.8) form. See additional retrieval query parameters (Section 3.7). OPTIONAL.
- excludedAttributes A multi-valued list of strings indicating the names of resource attributes to be removed from the default set of attributes to return. This parameter SHALL have no effect on attributes whose schema "returned" setting is "always" see Server Schema [I-D.ietf-scim-core-schema]. Attribute names MUST be in standard attribute notation (Section 3.8) form. See additional retrieval query parameters (Section 3.7). OPTIONAL.

CIFER Framework

5.2.4. Query String Options

2. Requests for large resources, e.g. group membership, may specify paging parameters

- sortField:
- sortOrder: 'a*', 'd*', case insensitive
- pageLimit: number of entries per page, zero means no paging
- Mutually exclusive page start:
  - pageOffset: page number, starting with zero, OR
  - pageStartValue: first value returned will be the one AFTER this value
- The service may provide default paging parameters, and may reduce the requested limits.

SCIM API

3.2.2.4 Pagination

Pagination parameters can be used together to "page through" large numbers of resources so as not to overwhelm the client or service provider. Pagination is not session based hence clients SHOULD never assume repeatable results. For example, a request for a list of 10 resources beginning with a startIndex of 1 may return different results when repeated as a resource in the original result could be deleted or new ones could be added in-between requests. Pagination parameters and general behavior are derived from the OpenSearch Protocol [OpenSearch].
CIFER Framework

5.2.4. Query String Options

3. Boolean parameters are generously interpreted

- true|false, or t|f or yes|no, or y|n or 1|0, case insensitive

SCIM API

3.2.2.2. Filtering

Filtering is OPTIONAL.

```
compValue = false / null / true / number / string
          ; rules from JSON (RFC7159)
```

NOTE: SCIM filtering options are as rich or richer than LDAP filter options. See section 3.2.2.2 for full details
CIFER Framework

5.2.4. Query String Options

4. A client may request a pretty-printed response with the query string, "indent=true"
SCIM API
NOTE: SCIM does not support client requests for pretty-printed responses.
CIFER Framework

5.2.4. Query String Options

5. Unexpected Query String parameters will be ignored.
SCIM API

NOTE: Primarily because SCIM works only with defined schema and extensions, unexpected query string parameters are not supported.
CIFER Framework

5.3. Request Methods

...
The framework does not specify these root and directory representations, except that they follow the general request and response guidelines.
SCIM API

3.2.2.1. Query Endpoints

A server MAY support searches against the server root (e.g. "/"). A search against a server root indicates that ALL resources within the server SHALL be included subject to filtering. A filter expression using "meta. resourceType" MAY be used to restrict results to one or more specific resource types (e.g. "User").
CIFER Framework

5.4.1 Response structure

Resources are returned along with metadata about the resource and metadata about the

response (see JSON examples that follow)

1. Resources are wrapped in an object which contains:

   1. resource: The actual resource for the request.

   2. resourceMetadata: metadata about the resource.

      fields as appropriate to the resource

lastModified: timestamp when the resource was modified
selfUri: URI to the current resource.
ETag, or equivalent, for resources that can be replaced, ...

3. responseMetadata: metadata about this particular response

responseTimestamp:
responseTime: time that the server took in processing this request
requestCommentary: freeform text about the request that the server processed (for debug)
httpStatusCode: response status as an HTTP status code.
serverVersion: major.minor server version number
paging parameters

pageLimit
pageOffset
sortField
pageOffsetValue
sortOrder



Review metadata approach

SCIM Core Schema

5.1 Common Schema Attributes

meta A complex attribute containing resource metadata. All sub-attributes are OPTIONAL

- resourceType: The name of the resource type of the resource. Attribute has mutability of "readOnly".
- created: The DateTime the resource was added to the service provider. The attribute MUST be a DateTime. Attribute has mutability of "readOnly".
- lastModified: The most recent DateTime the details of this resource were updated at the service provider. If this resource has never been modified since its initial creation, the value MUST be the same as the value of created. The attribute MUST be a DateTime. Attribute has mutability of "readOnly".
- location: The URI of the resource being returned. This value MUST be the same as the Location HTTP response header. Attribute has mutability of "readOnly".
- version: The version of the resource being returned. This value must be the same as the ETag HTTP response header. Attribute has mutability of "readOnly".
- attributes: The names of the attributes to remove from the resource during a PATCH operation.

CIFER Framework

5.4.2. Response Codes

1. 200: the resource was found
2. 201: the resource was created
3. 400: the request was not valid
4. 401: the client was unauthenticated
5. 403: an authenticated client was not authorized for the request
6. 404: a requested resource was not found
7. 409: (conflict)
    - PUT to an existing resource without an If-Match
8. 412: (precondition failed)
    - If-Match header did not match resource's ETag, or
    - If-Match header supplied but resource not found (add with an if-match), or
    - wildcard if-match not allowed
9. 500: server error



JF: We may want to include some content on a successful DELETE, then the response would be 200.

SCIM API

3.4 Deleting Resources

In response to a successful delete, the server SHALL respond with successful HTTP status 204 (No Content).

3.10 HTTP Error Responses

The SCIM Protocol uses the response status codes defined in HTTP Section 6 [RFC7231] to indicate operation success or failure. In addition to returning a HTTP response code implementers MUST return the errors in the body of the response in the client requested format containing the error response and, per the HTTP specification, human-readable explanations. Error responses are identified using the following "schema" URI: "urn:scim:api:messages:2.0:Error". The following attributes are defined inclusion in a SCIM error response using a JSON body

NOTE: See this section of the SCIM API specification for details on the meanings of specific error codes.