SCIM
Finalizing API / Core-Schema

SCIM WG

IETF90

July 23, 2014

Agenda

- SCIM API
  - Name / Terminology / Authenication
  - Missing Values & Mutability – PATCH/POST/PUT
  - DELETE – tombstoning issues
  - Bulk
  - Error Detail – extensions
- IANA Discussion
- Other Items
- Next Steps…

API / Protocol

- 07 uses API and protocol, needs to be made consistent
  - API is most often associated with programming implementations
  - Recommend:
    - Use SCIM Protocol or SCIM HTTP Protocol where appropriate
    - Leave document name unchanged

Terminology

- The term Identity Provider can be confusing
  - is this a client in SCIM?
  - Recommend:
    - Clarify in terminology section
    - Ensure drafts consistent around "client" and "service provider"
    - Avoid use of "identity provider"

Authentication

- Proposed new text:

```
2.  Authentication and Authorization

   Because SCIM builds on the HTTP protocol, it does not define a scheme
   for authentication and authorization.  SCIM depends on standard HTTP
   authentication schemes.  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 3.1 of [RFC7235].

   All examples show an OAuth2 bearer token [RFC6750] (though it is not
   required) ; 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.
```

PUT

- Issue: Missing Values
  - New text in 07

```
Attributes whose mutability is "readWrite" that are omitted from the
request body MAY be assumed to be not asserted by the client.  The
service provider MAY assume any existing values are to be cleared or
the service provider MAY assign a default value to the final resource
representation.  Service providers MAY take into account whether a
client has access to, or understands, all of the resource's
attributes when deciding whether non-asserted attributes SHALL be
removed or defaulted.  Clients that would like to override a server
defaults, MAY specify "null" for a single-valued attribute or an
empty array "[]" for a multi-valued attribute to clear all values.
```

  - Discuss/Confirm

PUT/POST/PATCH

- Review Attribute Mutability

DELETE

- Issue: To strict for tombstoning

  Clients request resource removal via DELETE. Service providers MAY choose not to permanently delete the resource, but MUST return a 404 error code for all operations associated with the previously deleted Id. Service providers MUST also omit the resource from future query results. **In addition the service provider MUST not consider the deleted resource in conflict calculation. For example if a User resource is deleted, a CREATE request for a User resource with the same userName as the previously deleted resource should not fail with a 409 error due to userName conflict.**

  – Recommend: Remove last 2 sentences

Bulk

- Issue: Unordered operations

```
There can be more then one operation per resource in each bulk job. The Service client MUST
take notice of the unordered structure of JSON and the service provider can process
operations in any order. For example, if the Service client sends two PUT operations in one
request, the outcome is non-deterministic.
```

  - Recommend: drop this paragraph
- Issue: Implementation
  - Are there async requirements as #ops increases?
  - Lack of practical implementation experience
  - Option: Break Bulk into separate WG draft and hold back WGLC

SCIM Example Error Response

```
HTTP/1.1 400 BAD REQUEST

 {
   "schemas": ["urn:scim:api:messages:2.0:Error"],
   "scimType":"mutability"
   "detail":"Attribute 'id' is readOnly",
   "status":"400"
 }
```

SCIM Detail Errors

- invalidFilter
- tooMany – too many results to calculate
- uniqueness – value in use or reserved
- mutability – attempt to modify an readOnly or immutable attribute
- invalidSyntax – request body syntax error
- invalidPath – the path specified is invalid for the operation
- noTarget – no target matched for the operation
- invalidValue – The value was not compatible with the operation (e.g. patch operation) or attribute type
- invalidVers – Protocol version not supported (e.g. v1)

Errors

- Issue: Whether or not to allow extensions
  - For new WG drafts to extend
  - For site specific high-level signals
    - (e.g. account quota exceeded, account is past due)

  - Recommend:
    - Make scimType a URI and use IANA to register errors
      - Allows new SCIM drafts to register new errors
    - For site specific: nothing
      - Sites can add their own attributes to the error detail
      - SCIM spec allows additional attributes

IANA Comments

- The current draft text was attempt to reflect what was in the spec and common usage.
  - Some amount of normalization was attempted
  - On reflection, the "formal" namespace may require simplification
  - For example, how would a 3rd party org define new schema. Would sub-name hold the org name? Would they use an entirely new namespace? Does it matter?
  - The order of some items might be questionable
  - There is confusion about urn's used for SCIM API Messages vs. SCIM data (is this an issue)
  - Discussion about allowing URIs

SCIM Core Schema

- IANA Schema URN Namespace

```
The Namespace Specific String (NSS) of all URNs that use the
     "scim" NID shall have the following structure:

  urn:scim:{type}:{name}{:subName}:{version}{:className}{:resourceType}

     The keywords have the following meaning:

     type
        An entity type (e.g. "schemas", "api", or "param" ).

     name
        A required US-ASCII string that conforms to the URN syntax
        requirements (see [RFC2141] ) and defines a major namespace of
        object used within SCIM (e.g. "core", "extension" ).  The name
        "extension" MUST be used when the registered schema it refers
        to is intended to be used as an extension to another schema.


        An optional US-ASCII string that conforms to the URN syntax
        requirements (see [RFC2141] ) and defines a sub-class of object
        used within SCIM (e.g. "enterprise", "extension" ).

     version
        The first SCIM API version number where the URN is valid (e.g.
        "2.0" ).
```

Schema URNs

- Core-Schema
  - urn:scim:schemas:core:2.0:User
  - urn:scim:schemas:extension:enterprise:2.0:User
  - urn:scim:schemas:core:2.0:Group
- API
  - urn:scim:api:messages:2.0
    - ListResponse, SearchRequest, PatchOp, BulkRequest, BulkResponse, Error
- Other (made-up) Examples
  - urn:scim:schemas:core:ExampleCo:2.0:app:Client
  - urn:scim:schemas:core:oAuth:2.0:Client
  - urn:scim:schemas:extension:ExampleCo:2.0:crm:socUser
  - urn:scim:schemas:extension:ExampleCo:2.0:crm:User

IANA Namespace

- Some Options
  - 1. Massage and re-work order of params with an idea to simplify (breaking – changes to urns)
  - 2. Leave as-is (avoid change even if awkward)
  - 3. Overall re-work (uri vs. urn's)
  - 4. Other?

Other Items?

Next Steps