

Schema for Users, Groups and SoRPeople for the BUS

Attributes Common Across All Three Resource Types

From SCIM Core Schema ([RFC 7643](#))

id

A unique identifier for a SCIM resource as defined by the service provider. Each representation of the resource MUST include a non-empty "id" value. This identifier MUST be unique across the SCIM 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 NOT be specified by the client. The string "bulkId" is a reserved keyword and MUST NOT be used within any unique identifier value. The attribute characteristics are "caseExact" as "true", a mutability of "readOnly", and a "returned" characteristic of "always". See [Section 9](#) for additional considerations regarding privacy.

externalId

A String that is an identifier for the resource as defined by the provisioning client. The "externalId" may simplify identification of a resource between the provisioning client and the service provider by allowing the client to use a filter to locate the resource with an identifier from the provisioning domain, obviating the need to store a local mapping between the provisioning domain's identifier of the resource and the identifier used by the service provider. Each resource MAY include a non-empty "externalId" value. The value of the "externalId" attribute is always issued by the provisioning client and MUST NOT be specified by the service provider. The service provider MUST always interpret the externalId as scoped to the provisioning domain. While the server does not enforce uniqueness, it is assumed that the value's uniqueness is controlled by the client setting the value.

[RFC 7644 SCIM Protocol: Section 3.4.2.5. Attributes](#)

The following attributes control which attributes SHALL be returned with a returned resource. SCIM clients MAY use one of these two OPTIONAL parameters, which MUST be supported by SCIM service providers:

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 3.10](#)) form. See [Section 3.9](#) for additional retrieval query parameters.

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 [Sections 2.2](#) and [7](#) of [\[RFC7643\]](#)). Attribute names MUST be in standard attribute notation ([Section 3.10](#)) form. See [Section 3.9](#) for additional retrieval query parameters.

Attributes Specific to SCIM Users ([RFC 7643](#))

[4.1.1. Singular Attributes](#)

[4.1.2. Multi-Valued Attributes](#)

Attributes Specific to SCIM Groups ([RFC 7643](#))

[4.2. "Group" Resource Schema](#)

testbed.tier.internet2.edu/secure as System of Record

--	--

| "Right now we capture: givenName, surname, initials (which I label as "middle name", mail, cn, uid, eppn, and displayName."

Here's the mapping from your attributes to their SCIM User Schema representation:

givenName, surname, initials (middleName), displayName (formatted name), uid, and eppn:

```
"uid": "userName":"bjensen",
"name": {"formatted":"Ms. Barbara J Jensen III"
"familyName":"Jensen",
"givenName":"Barbara"
"middleName": "Jane"
"displayName": "Babs Jensen" },
mail: "emails":[
  {"value":"bjensen@example.com"}
eppn: "externalId":"bjensen",
```

So, the JSON-formatted body of the POST would look like this:

```
{ "schemas":["urn:ietf:params:scim:schemas:core:2.0:User"],
  "userName":"bjensen",
  "externalId":"bjensen",
  "name": {
    "formatted":"Ms. Barbara J Jensen III"
    "familyName":"Jensen",
    "givenName":"Barbara",
    "middleName": "Jane",
    "displayName": "Babs Jensen"},
  "emails":[
    {"value":"bjensen@example.com"}
  ]
}
```

--Keith

From: Jim Jokl <jaj@virginia.edu>
Date: Friday, July 1, 2016 at 15:49
To: Keith Hazelton <keith.hazelton@wisc.edu>
Subject: RE: API Call

Looks simple enough.

Right now we capture: givenName, surname, initials (which I label as "middle name", mail, cn, uid, eppn, and displayName.

I overload "description" and use it to hold the creator's real home institution eppn

- so you can't have that - but I'm happy to add whatever else might be useful.

I use the home site EPPN for access control (i.e., you can only edit your own records)

I also have a password available or you can simply use Kerberos.

I don't store passwords in LDAP but could do that if you need it for some reason.

Jim

From: Keith Hazelton [mailto:keith.hazelton@wisc.edu]

Sent: Friday, July 1, 2016 4:40 PM

To: Jokl, James A. (Jim) (jaj) <jaj@virginia.edu>

Subject: Re: API Call

Let's start with this example straight out of the SCIM Protocol RFC.

Remind me what attributes you will have to pass along and I'll add elements to the request body to represent them

Let me know if you have any questions. -Keith

RFC 7644 SCIM Protocol Specification September 2015

...

When the service provider successfully creates the new resource, an HTTP response SHALL be returned with HTTP status code 201 (Created). The response body SHOULD contain the service provider's representation of the newly created resource. The URI of the created resource SHALL include, in the HTTP "Location" header and the HTTP body, a JSON representation [RFC7159] with the attribute "meta.location". Since the server is free to alter and/or ignore POSTed content, returning the full representation can be useful to the client, enabling it to correlate the client's and server's views of the new resource.

In the following example, a client sends a POST request containing a "User" to the "/Users" endpoint.

POST /Users HTTP/1.1

Host: example.com

Accept: application/scim+json

Content-Type: application/scim+json

Authorization: Bearer h480djs93hd8

Content-Length: ...

```
{
  "schemas":["urn:ietf:params:scim:schemas:core:2.0:User"],
  "userName":"bjensen",
  "externalId":"bjensen",
  "name":{"
    "formatted":"Ms. Barbara J Jensen III",
    "familyName":"Jensen",
    "givenName":"Barbara"
  }
}
```

In response to the example request above, the server signals a successful creation with an HTTP status code 201 (Created) and returns a representation of the resource created:

HTTP/1.1 201 Created

Content-Type: application/scim+json

Location:

<https://example.com/v2/Users/2819c223-7f76-453a-919d-413861904646>

ETag: W/"e180ee84f0671b1"

```
{
  "schemas":["urn:ietf:params:scim:schemas:core:2.0:User"],
  "id":"2819c223-7f76-453a-919d-413861904646",
  "externalId":"bjensen",
  "meta":{"
    "resourceType":"User",
    "created":"2011-08-01T21:32:44.882Z",
    "lastModified":"2011-08-01T21:32:44.882Z",
    "location":
"https://example.com/v2/Users/2819c223-7f76-453a-919d-413861904646",
    "version":"W\/"e180ee84f0671b1\" "
  },
  "name":{"
    "formatted":"Ms. Barbara J Jensen III",
    "familyName":"Jensen",
    "givenName":"Barbara"
  },
  "userName":"bjensen"
}
```

