

Internet Engineering Task Force (IETF)
Request for Comments: 7643
Category: Standards Track
ISSN: 2070-1721

P. Hunt, Ed.
Oracle
K. Grizzle
SailPoint
E. Wahlstroem
Nexus Technology
C. Mortimore
Salesforce
September 2015

System for Cross-domain Identity Management: Core Schema

Abstract

The System for Cross-domain Identity Management (SCIM) specifications are designed to make identity management in cloud-based applications and services easier. The specification suite builds upon experience with existing schemas and deployments, placing specific emphasis on simplicity of development and integration, while applying existing authentication, authorization, and privacy models. Its intent is to reduce the cost and complexity of user management operations by providing a common user schema and extension model as well as binding documents to provide patterns for exchanging this schema using HTTP.

This document provides a platform-neutral schema and extension model for representing users and groups and other resource types in JSON format. This schema is intended for exchange and use with cloud service providers.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at <http://www.rfc-editor.org/info/rfc7643>.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction and Overview	3
1.1. Requirements Notation and Conventions	4
1.2. Definitions	5
2. SCIM Schema	6
2.1. Attributes	7
2.2. Attribute Characteristics	8
2.3. Attribute Data Types	8
2.3.1. String	9
2.3.2. Boolean	9
2.3.3. Decimal	10
2.3.4. Integer	10
2.3.5. DateTime	10
2.3.6. Binary	10
2.3.7. Reference	10
2.3.8. Complex	11
2.4. Multi-Valued Attributes	11
2.5. Unassigned and Null Values	13
3. SCIM Resources	13
3.1. Common Attributes	16
3.2. Defining New Resource Types	18
3.3. Attribute Extensions to Resources	18
4. SCIM Core Resources and Extensions	19
4.1. "User" Resource Schema	19
4.1.1. Singular Attributes	19
4.1.2. Multi-Valued Attributes	23
4.2. "Group" Resource Schema	25
4.3. Enterprise User Schema Extension	26
5. Service Provider Configuration Schema	27
6. ResourceType Schema	29
7. Schema Definition	30

- 8. JSON Representation34
 - 8.1. Minimal User Representation34
 - 8.2. Full User Representation35
 - 8.3. Enterprise User Extension Representation39
 - 8.4. Group Representation43
 - 8.5. Service Provider Configuration Representation44
 - 8.6. Resource Type Representation46
 - 8.7. Schema Representation47
 - 8.7.1. Resource Schema Representation47
 - 8.7.2. Service Provider Schema Representation74
- 9. Security Considerations92
 - 9.1. Protocol92
 - 9.2. Passwords and Other Sensitive Security Data92
 - 9.3. Privacy92
- 10. IANA Considerations94
 - 10.1. Registration of SCIM URN Sub-namespace and SCIM Registry94
 - 10.2. URN Sub-namespace for SCIM94
 - 10.2.1. Specification Template95
 - 10.3. Registering SCIM Schemas97
 - 10.3.1. Registration Procedure97
 - 10.3.2. Schema Registration Template98
 - 10.4. Initial SCIM Schema Registry99
- 11. References100
 - 11.1. Normative References100
 - 11.2. Informative References101
- Acknowledgements103
- Authors' Addresses104

1. Introduction and Overview

While there are existing standards for describing and exchanging user information, many of these standards can be difficult to implement and/or use; e.g., their wire protocols do not easily traverse firewalls and/or are not easily layered onto existing web protocols. As a result, many cloud providers implement non-standardized protocols for managing users within their services. This increases both the cost and complexity associated with organizations adopting products and services from multiple cloud providers, as they must perform redundant integration development. Similarly, cloud service providers seeking to interoperate with multiple application marketplaces or cloud identity providers would require pairwise integration.

SCIM seeks to simplify this problem through an easily implemented specification suite that provides a common user schema and extension model, as well as a SCIM protocol document that defines exchanging this schema via an HTTP-based protocol [RFC7644]. The SCIM

specifications draw design input and feedback from existing identity-related protocols and schemas from a wide variety of sources including, but not limited to, existing services exposed by cloud providers, PortableContacts [PortableContacts], vCards [RFC6350], and Lightweight Directory Access Protocol (LDAP) directory services [RFC4512].

The SCIM protocol is an application-level protocol for provisioning and managing identity data specified through SCIM schemas. The protocol supports creation, modification, retrieval, and discovery of core identity resources such as Users and Groups, using a subset of the HTTP methods (GET for retrieval of resources; POST for creation, searching, and bulk modification; PUT for attribute replacement within resources; PATCH for partial update of attributes; and DELETE for removing resources).

While the SCIM protocol and core schema specifications are intended to cover point-to-point scenarios, implementers and deployers should consider multi-hop and multi-party scenarios such as a service provider acting as a general profile service for in-domain applications (e.g., a directory), as well as scenarios where a service provider in turn passes information to a third-party service provider by acting as either a SCIM client or a SCIM service provider. Implementers and deployers should carefully consider their service level agreements and privacy agreements when distributing or propagating personal information (see Section 9.3).

This document provides a JSON-based schema and extension model for representing users and groups, as well as service provider configuration. This schema is intended for exchange and use with cloud service providers and other cross-domain scenarios.

1.1. Requirements Notation and Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

The key words "REQUIRED" and "OPTIONAL" are used throughout this document to indicate whether an attribute or schema element is required or optional. These key words may be used alone (e.g., "REQUIRED.") or in a sentence. If not specified, an attribute is considered to be optional.

The word "DEFAULT" as used in Section 7 indicates that a "keyword" value for an attribute characteristic is the default behavior.

Throughout this document, values are quoted to indicate that they are to be taken literally. When using these values in protocol messages, the quotes MUST NOT be used as part of the value.

Throughout this document, figures may contain spaces and extra line wrapping to improve readability and accommodate space limitations. Similarly, some URIs contained within examples have been shortened for space and readability reasons.

1.2. Definitions

Service Provider

An HTTP web application that provides identity information via the SCIM protocol.

Client

A website or application that uses the SCIM protocol to manage identity data maintained by the service provider. The client initiates SCIM HTTP requests to a target service provider.

Provisioning Domain

A provisioning domain is an administrative domain external to the domain of a service provider for legal or technical reasons. For example, a SCIM client in an enterprise (provisioning client) communicates with a SCIM service provider that is owned or controlled by a different legal entity.

Resource Type

A type of a resource that is managed by a service provider. The resource type defines the resource name, endpoint URL, schemas, and other metadata that indicate where a resource is managed and how it is composed, e.g., "User" or "Group".

Resource

An artifact that is managed by a service provider and that contains one or more attributes, e.g., "User" or "Group".

Endpoint

An endpoint for a service provider is a defined base path relative to the service provider's Base URI (see Section 1.3 of [RFC7644]), over which SCIM operations may be performed against SCIM resources. For example, assuming that the service provider's Base URI is "https://example.com/", "User" resources may be accessed at the "https://example.com/Users" or "https://example.com/v2/Users" endpoint (see Section 3.13 of [RFC7644] for details regarding protocol versioning, e.g., 'v2'). Service provider schemas MAY be returned from the "/Schemas" endpoint.

Schema

A collection of attribute definitions that describe the contents of an entire or partial resource, e.g., "urn:ietf:params:scim:schemas:core:2.0:User". The attribute definitions specify the name of the attribute, and metadata such as type (e.g., string, binary), cardinality (singular, multi, complex), mutability, and returnability.

Singular Attribute

A resource attribute that contains 0..1 values, e.g., "displayName".

Multi-valued Attribute

A resource attribute that contains 0..n values, e.g., "emails".

Simple Attribute

A singular or multi-valued attribute whose value is a primitive, e.g., "String". A simple attribute MUST NOT contain sub-attributes.

Complex Attribute

A singular or multi-valued attribute whose value is a composition of one or more simple attributes; e.g., "addresses" has the sub-attributes "streetAddress", "locality", "postalCode", and "country".

Sub-Attribute

A simple attribute that is contained within a complex attribute.

2. SCIM Schema

A SCIM server provides a set of resources, the allowable contents of which are defined by a set of schema URIs and a resource type. SCIM's schema is not a document-centric one such as with [XML-Schema]. Instead, SCIM's support of schema is attribute based, where each attribute may have different type, mutability, cardinality, or returnability. Validation of documents and messages is always performed by an intended receiver, as specified by the SCIM specifications. Validation is performed by the receiver in the context of a SCIM protocol request (see [RFC7644]). For example, a SCIM service provider, upon receiving a request to replace an existing resource with a replacement JSON object, evaluates each asserted attribute based on its characteristics as defined in the relevant schema (e.g., mutability) and decides which attributes may be replaced or ignored.

This specification provides a minimal core schema for representing users and groups (resources), encompassing common attributes found in many existing deployments and schemas. In addition to the minimal core schema, this document also specifies a standardized means by which service providers may extend schemas to define new resources and attributes in both standardized and service-provider-specific cases.

Resources are categorized into common resource types such as "User" or "Group". Collections of resources of the same type are usually contained within the same "container" ("folder") endpoint.

2.1. Attributes

A resource is a collection of attributes identified by one or more schemas. Minimally, an attribute consists of the attribute name and at least one simple or complex value, either of which may be multi-valued. For each attribute, a SCIM schema defines the data type, plurality, mutability, and other distinguishing features of an attribute.

Attribute names are case insensitive and are often "camel-cased" (e.g., "camelCase"). SCIM resources are represented in JSON [RFC7159] format and MUST specify schema via the "schemas" attribute per Section 3.

Attribute names MUST conform to the following ABNF rules:

```
ATTRNAME    = ALPHA *(nameChar)
nameChar    = "$" / "-" / "_" / DIGIT / ALPHA
```

Figure 1: ABNF for Attribute Names

The above rules (and other rules in this specification) use the "Core Rules" from ABNF; see Appendix B of [RFC5234]. Unless otherwise specified in this document, all ABNF strings are case insensitive and the character set for these strings is US-ASCII. For example, all attribute names defined by the above rule are case insensitive.

When defining attribute names, it should be noted that the hyphen ("-") is not permitted in JavaScript attribute names (or in attribute names for some other languages). While there are no known issues within HTTP protocol and JSON notation, attribute names containing hyphens may need to be escaped when declaring corresponding names of JavaScript attributes.

2.2. Attribute Characteristics

All attributes have a set of characteristics that describe their type and handling by a service provider; full definitions may be found in Section 7. The characteristics include:

- o "required",
- o "canonicalValues",
- o "caseExact",
- o "mutability",
- o "returned",
- o "uniqueness", and
- o "referenceTypes".

If not otherwise stated in Section 7, SCIM attributes have the following characteristics:

- o "required" is "false" (i.e., not REQUIRED),
- o "canonicalValues": none assigned (for example, the "type" sub-attribute as described in Section 2.4),
- o "caseExact" is "false" (i.e., case-insensitive),
- o "mutability" is "readWrite" (i.e., modifiable),
- o "returned" is "default" (the attribute value is returned by default),
- o "uniqueness" is "none" (has no uniqueness enforced), and
- o "type" is "string" (Section 2.3.1).

2.3. Attribute Data Types

Attribute data types are derived from JSON [RFC7159]. The JSON format defines a limited set of data types; hence, where appropriate, alternate JSON representations derived from XML Schema [XML-Schema] are defined below. SCIM extensions SHOULD NOT introduce new data types.

Table 1 maps the following SCIM data types to their corresponding SCIM schema type and underlying JSON data type:

SCIM Data Type	SCIM Schema "type"	JSON Type
String	"string"	String per Section 7 of [RFC7159]
Boolean	"boolean"	Value per Section 3 of [RFC7159]
Decimal	"decimal"	Number per Section 6 of [RFC7159]
Integer	"integer"	Number per Section 6 of [RFC7159]
DateTime	"dateTime"	String per Section 7 of [RFC7159]
Binary	"binary"	Binary value base64 encoded per Section 4 of [RFC4648], or with URL and filename safe alphabet URL per Section 5 of [RFC4648] that is passed as a JSON string per Section 7 of [RFC7159]
Reference	"reference"	String per Section 7 of [RFC7159]
Complex	"complex"	Object per Section 4 of [RFC7159]

Table 1: SCIM Data Type to JSON Representation

2.3.1. String

A sequence of zero or more Unicode characters encoded using UTF-8 as per [RFC2277] and [RFC3629]. The JSON format is defined in Section 7 of [RFC7159]. An attribute with SCIM schema type "string" MAY specify a required data format. Additionally, when "canonicalValues" is specified, service providers MAY restrict accepted values to the specified values.

2.3.2. Boolean

The literal "true" or "false". The JSON format is defined in Section 3 of [RFC7159]. A boolean has no case sensitivity or uniqueness.

2.3.3. Decimal

A real number with at least one digit to the left and right of the period. The JSON format is defined in Section 6 of [RFC7159]. A decimal has no case sensitivity.

2.3.4. Integer

A whole number with no fractional digits or decimal. The JSON format is defined in Section 6 of [RFC7159], with the additional constraint that the value MUST NOT contain fractional or exponent parts. An integer has no case sensitivity.

2.3.5. 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 3.3.7 of [XML-Schema] and MUST include both a date and a time. A date time format has no case sensitivity or uniqueness.

Values represented in JSON format MUST conform to the XML constraints above and are represented as a JSON string per Section 7 of [RFC7159].

2.3.6. Binary

Arbitrary binary data. The attribute value MUST be base64 encoded as specified in Section 4 of [RFC4648]. In cases where a URL-safe encoding is required, the attribute definition MAY specify that base64 URL encoding be used as per Section 5 of [RFC4648]. Unless otherwise specified in the attribute definition, trailing padding characters MAY be omitted ("=").

In JSON representation, the encoded values are represented as a JSON string per Section 7 of [RFC7159]. A binary is case exact and has no uniqueness.

2.3.7. Reference

A URI for a resource. A resource MAY be a SCIM resource, an external link to a resource (e.g., a photo), or an identifier such as a URN. The value MUST be the absolute or relative URI of the target resource. Relative URIs should be resolved as specified in Section 5.2 of [RFC3986]. However, the base URI for relative URI resolution MUST include all URI components and path segments up to, but not including, the Endpoint URI (the SCIM service provider root

endpoint); e.g., the base URI for a request to "https://example.com/v2/Users/2819c223-7f76-453a-919d-413861904646" would be "https://example.com/v2/", and the relative URI for this resource would be "Users/2819c223-7f76-453a-919d-413861904646".

In JSON representation, the URI value is represented as a JSON string per Section 7 of [RFC7159]. A reference is case exact. A reference has a "referenceTypes" attribute that indicates what types of resources may be linked, as per Section 7 of this document.

A reference URI MUST be to an HTTP-addressable resource. An HTTP client performing a GET operation on a reference URI MUST receive the target resource or an appropriate HTTP response code. A SCIM service provider MAY choose to enforce referential integrity for reference types referring to SCIM resources.

By convention, a reference is commonly represented as a "\$ref" sub-attribute in complex or multi-valued attributes; however, this is OPTIONAL.

2.3.8. Complex

A singular or multi-valued attribute whose value is a composition of one or more simple attributes. The JSON format is defined in Section 4 of [RFC7159]. The order of the component attributes is not significant. Servers and clients MUST NOT require or expect attributes to be in any specific order when an object is either generated or analyzed. A complex attribute has no uniqueness or case sensitivity. A complex attribute MUST NOT contain sub-attributes that have sub-attributes (i.e., that are complex).

2.4. Multi-Valued Attributes

Multi-valued attributes contain a list of elements using the JSON array format defined in Section 5 of [RFC7159]. Elements can be either of the following:

- o primitive values, or
- o objects with a set of sub-attributes and values, using the JSON object format defined in Section 4 of [RFC7159], in which case they SHALL be considered to be complex attributes. As with complex attributes, the order of sub-attributes is not significant. The predefined sub-attributes listed in this section can be used with multi-valued attribute objects, but these sub-attributes MUST be used with the meanings defined here.

If not otherwise defined, the default set of sub-attributes for a multi-valued attribute is as follows:

type

A label indicating the attribute's function, e.g., "work" or "home".

primary

A Boolean value indicating the 'primary' or preferred attribute value for this attribute, e.g., the preferred mailing address or the primary email address. The primary attribute value "true" MUST appear no more than once. If not specified, the value of "primary" SHALL be assumed to be "false".

display

A human-readable name, primarily used for display purposes and having a mutability of "immutable".

value

The attribute's significant value, e.g., email address, phone number.

\$ref

The reference URI of a target resource, if the attribute is a reference. URIs are canonicalized per Section 6.2 of [RFC3986]. While the representation of a resource may vary in different SCIM protocol API versions (see Section 3.13 of [RFC7644]), URIs for SCIM resources with an API version SHALL be considered comparable to URIs without a version or with a different version. For example, "https://example.com/Users/12345" is equivalent to "https://example.com/v2/Users/12345".

When returning multi-valued attributes, service providers SHOULD canonicalize the value returned (e.g., by returning a value for the sub-attribute "type", such as "home" or "work") when appropriate (e.g., for email addresses and URLs).

Service providers MAY return element objects with the same "value" sub-attribute more than once with a different "type" sub-attribute (e.g., the same email address may be used for work and home) but SHOULD NOT return the same (type, value) combination more than once per attribute, as this complicates processing by the client.

When defining schema for multi-valued attributes, it is considered a good practice to provide a type attribute that MAY be used for the purpose of canonicalization of values. In the schema definition for an attribute, the service provider MAY define the recommended canonical values (see Section 7).

2.5. Unassigned and Null Values

Unassigned attributes, the null value, or an empty array (in the case of a multi-valued attribute) SHALL be considered to be equivalent in "state". Assigning an attribute with the value "null" or an empty array (in the case of multi-valued attributes) has the effect of making the attribute "unassigned". When a resource is expressed in JSON format, unassigned attributes, although they are defined in schema, MAY be omitted for compactness.

3. SCIM Resources

Each SCIM resource is a JSON object that has the following components:

Resource Type

Each resource (or JSON object) in SCIM has a resource type ("meta.resourceType"; see Section 3.1) that defines the resource's core attribute schema and any attribute extension schema, as well as the endpoint where objects of the same type may be found. More information about a resource MAY be found in its resource type definition (see Section 6).

"Schemas" Attribute

The "schemas" attribute is a REQUIRED attribute and is an array of Strings containing URIs that are used to indicate the namespaces of the SCIM schemas that define the attributes present in the current JSON structure. This attribute may be used by parsers to define the attributes present in the JSON structure that is the body to an HTTP request or response. Each String value must be a unique URI. All representations of SCIM schemas MUST include a non-empty array with value(s) of the URIs supported by that representation. The "schemas" attribute for a resource MUST only contain values defined as "schema" and "schemaExtensions" for the resource's defined "resourceType". Duplicate values MUST NOT be included. Value order is not specified and MUST NOT impact behavior.

Common Attributes

A resource's common attributes are those attributes that are part of every SCIM resource, regardless of the value of the "schemas" attribute present in a JSON body. These attributes are not defined in any particular schema but SHALL be assumed to be present in every resource, regardless of the value of the "schemas" attribute. See Section 3.1.

Core Attributes

A resource's core attributes are those attributes that sit at the top level of the JSON object together with the common attributes (such as the resource "id"). The list of valid attributes is specified by the resource's resource type "schema" attribute (see Section 6). This same value is also present in the resource's "schemas" attribute.

Extended Attributes

Extended schema attributes are specified by the resource's resource type "schemaExtensions" attribute (see Section 6). Unlike core attributes, extended attributes are kept in their own sub-attribute namespace identified by the schema extension URI. This avoids attribute name conflicts that may arise due to conflicts from separate schema extensions.

The following example "User" contains the common attributes "id" and "externalId", as well as the complex attribute "meta", which contains the sub-attribute "resourceType". The resource also contains core attributes "userName" and "name", as well as extended enterprise User attributes "employeeNumber" and "costCenter", which are contained in their own JSON substructure identified by their schema URI. Some values have been omitted (...), shortened, or spaced out for clarity.

```
{
  "schemas":
    ["urn:ietf:params:scim:schemas:core:2.0:User",
     "urn:ietf:params:scim:schemas:extension:enterprise:2.0:User"],

  "id": "2819c223-7f76-453a-413861904646",
  "externalId": "701984",

  "userName": "bjensen@example.com",
  "name": {
    "formatted": "Ms. Barbara J Jensen, III",
    "familyName": "Jensen",
    "givenName": "Barbara",
    "middleName": "Jane",
    "honorificPrefix": "Ms.",
    "honorificSuffix": "III"
  },
  ...

  "urn:ietf:params:scim:schemas:extension:enterprise:2.0:User": {
    "employeeNumber": "701984",
    "costCenter": "4130",
    ...
  },

  "meta": {
    "resourceType": "User",
    "created": "2010-01-23T04:56:22Z",
    "lastModified": "2011-05-13T04:42:34Z",
    "version": "W\/"3694e05e9dff591\"",
    "location":
      "https://example.com/v2/Users/2819c223-7f76-453a-413861904646"
  }
}
```

Figure 2: Example JSON Resource Structure

3.1. Common Attributes

Each SCIM resource (Users, Groups, etc.) includes the following common attributes. With the exception of the "ServiceProviderConfig" and "ResourceType" server discovery endpoints and their associated resources, these attributes MUST be defined for all resources, including any extended resource types. When accepted by a service provider (e.g., after a SCIM create), the attributes "id" and "meta" (and its associated sub-attributes) MUST be assigned values by the service provider. Common attributes are considered to be part of every base resource schema and do not use their own "schemas" URI.

For backward compatibility, some existing schema definitions MAY list common attributes as part of the schema. The attribute characteristics (see Section 2.2) listed here SHALL take precedence over older definitions that may be included in existing schemas.

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. See Section 9 for

additional considerations regarding privacy. This attribute has "caseExact" as "true" and a mutability of "readWrite". This attribute is OPTIONAL.

meta

A complex attribute containing resource metadata. All "meta" sub-attributes are assigned by the service provider (have a "mutability" of "readOnly"), and all of these sub-attributes have a "returned" characteristic of "default". This attribute SHALL be ignored when provided by clients. "meta" contains the following sub-attributes:

resourceType The name of the resource type of the resource. This attribute has a mutability of "readOnly" and "caseExact" as "true".

created The "DateTime" that the resource was added to the service provider. This attribute MUST be a DateTime.

lastModified The most recent DateTime that 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".

location The URI of the resource being returned. This value MUST be the same as the "Content-Location" HTTP response header (see Section 3.1.4.2 of [RFC7231]).

version The version of the resource being returned. This value must be the same as the entity-tag (ETag) HTTP response header (see Sections 2.1 and 2.3 of [RFC7232]). This attribute has "caseExact" as "true". Service provider support for this attribute is optional and subject to the service provider's support for versioning (see Section 3.14 of [RFC7644]). If a service provider provides "version" (entity-tag) for a representation and the generation of that entity-tag does not satisfy all of the characteristics of a strong validator (see Section 2.1 of [RFC7232]), then the origin server MUST mark the "version" (entity-tag) as weak by prefixing its opaque value with "W/" (case sensitive).

3.2. Defining New Resource Types

SCIM may be extended to define new classes of resources by defining a resource type. Each resource type defines the name, endpoint, base schema (the attributes), and any schema extensions registered for use with the resource type. In order to offer new types of resources, a service provider defines the new resource type as specified in Section 6 and defines a schema representation (see Section 8.7).

3.3. Attribute Extensions to Resources

SCIM allows resource types to have extensions in addition to their core schema. This is similar to how "objectClasses" are used in LDAP [RFC4512]. However, unlike LDAP, there is no inheritance model; all extensions are additive (similar to the LDAP auxiliary object class). Each value in the "schemas" attribute indicates additive schema that MAY exist in a SCIM resource representation. The "schemas" attribute MUST contain at least one value, which SHALL be the base schema for the resource. The "schemas" attribute MAY contain additional values indicating extended schemas that are in use. Schema extensions SHOULD avoid redefining any attributes defined in this specification and SHOULD follow conventions defined in this specification. Except for the base object schema, the schema extension URI SHALL be used as a JSON container to distinguish attributes belonging to the extension namespace from base schema attributes. See Figure 5, which is an example of the JSON representation of an enterprise User and is also an example of a User with extended schema.

In order to determine which URI value in the "schemas" attribute is the base schema and which is an extended schema for any given resource, the resource's "resourceType" attribute value MAY be used to retrieve the resource's "ResourceType" schema (see Section 6). See the "ResourceType" representation in Figure 8 for an example.

4. SCIM Core Resources and Extensions

This section defines the default resource schemas present in a SCIM server. SCIM is not exclusive to these resources and may be extended to support other resource types (see Section 3.2).

4.1. "User" Resource Schema

SCIM provides a resource type for "User" resources. The core schema for "User" is identified using the following schema URI: "urn:ietf:params:scim:schemas:core:2.0:User". The following attributes are defined in addition to the core schema attributes:

4.1.1. Singular Attributes

userName

A service provider's unique identifier for the user, typically used by the user to directly authenticate to the service provider. Often displayed to the user as their unique identifier within the system (as opposed to "id" or "externalId", which are generally opaque and not user-friendly identifiers). Each User MUST include a non-empty userName value. This identifier MUST be unique across the service provider's entire set of Users. This attribute is REQUIRED and is case insensitive.

name

The components of the user's name. Service providers MAY return just the full name as a single string in the formatted sub-attribute, or they MAY return just the individual component attributes using the other sub-attributes, or they MAY return both. If both variants are returned, they SHOULD be describing the same name, with the formatted name indicating how the component attributes should be combined.

formatted The full name, including all middle names, titles, and suffixes as appropriate, formatted for display (e.g., "Ms. Barbara Jane Jensen, III").

familyName The family name of the User, or last name in most Western languages (e.g., "Jensen" given the full name "Ms. Barbara Jane Jensen, III").

givenName The given name of the User, or first name in most Western languages (e.g., "Barbara" given the full name "Ms. Barbara Jane Jensen, III").

middleName The middle name(s) of the User (e.g., "Jane" given the full name "Ms. Barbara Jane Jensen, III").

honorificPrefix The honorific prefix(es) of the User, or title in most Western languages (e.g., "Ms." given the full name "Ms. Barbara Jane Jensen, III").

honorificSuffix The honorific suffix(es) of the User, or suffix in most Western languages (e.g., "III" given the full name "Ms. Barbara Jane Jensen, III").

displayName

The name of the user, suitable for display to end-users. Each user returned MAY include a non-empty displayName value. The name SHOULD be the full name of the User being described, if known (e.g., "Babs Jensen" or "Ms. Barbara J Jensen, III") but MAY be a username or handle, if that is all that is available (e.g., "bjensen"). The value provided SHOULD be the primary textual label by which this User is normally displayed by the service provider when presenting it to end-users.

nickName

The casual way to address the user in real life, e.g., "Bob" or "Bobby" instead of "Robert". This attribute SHOULD NOT be used to represent a User's username (e.g., bjensen or mpepperidge).

profileUrl

A URI that is a uniform resource locator (as defined in Section 1.1.3 of [RFC3986]) and that points to a location representing the user's online profile (e.g., a web page). URIs are canonicalized per Section 6.2 of [RFC3986].

title

The user's title, such as "Vice President".

userType

Used to identify the relationship between the organization and the user. Typical values used might be "Contractor", "Employee", "Intern", "Temp", "External", and "Unknown", but any value may be used.

preferredLanguage

Indicates the user's preferred written or spoken languages and is generally used for selecting a localized user interface. The value indicates the set of natural languages that are preferred. The format of the value is the same as the HTTP Accept-Language header field (not including "Accept-Language:") and is specified in Section 5.3.5 of [RFC7231]. The intent of this value is to enable cloud applications to perform matching of language tags [RFC4647] to the user's language preferences, regardless of what may be indicated by a user agent (which might be shared), or in an

interaction that does not involve a user (such as in a delegated OAuth 2.0 [RFC6749] style interaction) where normal HTTP Accept-Language header negotiation cannot take place.

locale

Used to indicate the User's default location for purposes of localizing such items as currency, date time format, or numerical representations. A valid value is a language tag as defined in [RFC5646]. Computer languages are explicitly excluded.

A language tag is a sequence of one or more case-insensitive sub-tags, each separated by a hyphen character ("-"; %x2D). For backward compatibility, servers MAY accept tags separated by an underscore character ("_"; %x5F). In most cases, a language tag consists of a primary language sub-tag that identifies a broad family of related languages (e.g., "en" = English) and that is optionally followed by a series of sub-tags that refine or narrow that language's range (e.g., "en-CA" = the variety of English as communicated in Canada). Whitespace is not allowed within a language tag. Example tags include:

fr, en-US, es-419, az-Arab, x-pig-latin, man-Nkoo-GN

See [RFC5646] for further information.

timezone

The User's time zone, in IANA Time Zone database format [RFC6557], also known as the "Olson" time zone database format [Olson-TZ] (e.g., "America/Los_Angeles").

active

A Boolean value indicating the user's administrative status. The definitive meaning of this attribute is determined by the service provider. As a typical example, a value of true implies that the user is able to log in, while a value of false implies that the user's account has been suspended.

`password`

This attribute is intended to be used as a means to set, replace, or compare (i.e., filter for equality) a password. The cleartext value or the hashed value of a password SHALL NOT be returnable by a service provider. If a service provider holds the value locally, the value SHOULD be hashed. When a password is set or changed by the client, the cleartext password SHOULD be processed by the service provider as follows:

- * Prepare the cleartext value for international language comparison. See Section 7.8 of [RFC7644].
- * Validate the value against server password policy. Note: The definition and enforcement of password policy are beyond the scope of this document.
- * Ensure that the value is encrypted (e.g., hashed). See Section 9.2 for acceptable hashing and encryption handling when storing or persisting for provisioning workflow reasons.

A service provider that immediately passes the cleartext value on to another system or programming interface MUST pass the value directly over a secured connection (e.g., Transport Layer Security (TLS)). If the value needs to be temporarily persisted for a period of time (e.g., because of a workflow) before provisioning, then the value MUST be protected by some method, such as encryption.

Testing for an equality match MAY be supported if there is an existing stored hashed value. When testing for equality, the service provider:

- * Prepares the filter value for international language comparison. See Section 7.8 of [RFC7644].
- * Generates the salted hash of the filter value and tests for a match with the locally held value.

The mutability of the password attribute is "writeOnly", indicating that the value MUST NOT be returned by a service provider in any form (the attribute characteristic "returned" is "never").

4.1.2. Multi-Valued Attributes

The following multi-valued attributes are defined.

emails

Email addresses for the User. The value SHOULD be specified according to [RFC5321]. Service providers SHOULD canonicalize the value according to [RFC5321], e.g., "bjensen@example.com" instead of "bjensen@EXAMPLE.COM". The "display" sub-attribute MAY be used to return the canonicalized representation of the email value. The "type" sub-attribute is used to provide a classification meaningful to the (human) user. The user interface should encourage the use of basic values of "work", "home", and "other" and MAY allow additional type values to be used at the discretion of SCIM clients.

phoneNumbers

Phone numbers for the user. The value SHOULD be specified according to the format defined in [RFC3966], e.g., 'tel:+1-201-555-0123'. Service providers SHOULD canonicalize the value according to [RFC3966] format, when appropriate. The "display" sub-attribute MAY be used to return the canonicalized representation of the phone number value. The sub-attribute "type" often has typical values of "work", "home", "mobile", "fax", "pager", and "other" and MAY allow more types to be defined by the SCIM clients.

ims

Instant messaging address for the user. No official canonicalization rules exist for all instant messaging addresses, but service providers SHOULD, when appropriate, remove all whitespace and convert the address to lowercase. The "type" sub-attribute SHOULD take one of the following values: "aim", "gtalk", "icq", "xmpp", "msn", "skype", "qq", "yahoo", or "other" (representing currently popular IM services at the time of this writing). Service providers MAY add further values if new IM services are introduced and MAY specify more detailed canonicalization rules for each possible value.

photos

A URI that is a uniform resource locator (as defined in Section 1.1.3 of [RFC3986]) that points to a resource location representing the user's image. The resource MUST be a file (e.g., a GIF, JPEG, or PNG image file) rather than a web page containing an image. Service providers MAY return the same image in different sizes, although it is recognized that no standard for describing images of various sizes currently exists. Note that this attribute SHOULD NOT be used to send down arbitrary photos

taken by this user; instead, profile photos of the user that are suitable for display when describing the user should be sent. Instead of the standard canonical values for type, this attribute defines the following canonical values to represent popular photo sizes: "photo" and "thumbnail".

addresses

A physical mailing address for this user. Canonical type values of "work", "home", and "other". This attribute is a complex type with the following sub-attributes. All sub-attributes are OPTIONAL.

formatted The full mailing address, formatted for display or use with a mailing label. This attribute MAY contain newlines.

streetAddress The full street address component, which may include house number, street name, P.O. box, and multi-line extended street address information. This attribute MAY contain newlines.

locality The city or locality component.

region The state or region component.

postalCode The zip code or postal code component.

country The country name component. When specified, the value MUST be in ISO 3166-1 "alpha-2" code format [ISO3166]; e.g., the United States and Sweden are "US" and "SE", respectively.

groups

A list of groups to which the user belongs, either through direct membership, through nested groups, or dynamically calculated. The values are meant to enable expression of common group-based or role-based access control models, although no explicit authorization model is defined. It is intended that the semantics of group membership and any behavior or authorization granted as a result of membership are defined by the service provider. The canonical types "direct" and "indirect" are defined to describe how the group membership was derived. Direct group membership indicates that the user is directly associated with the group and SHOULD indicate that clients may modify membership through the "Group" resource. Indirect membership indicates that user membership is transitive or dynamic and implies that clients cannot modify indirect group membership through the "Group" resource but MAY modify direct group membership through the "Group" resource, which may influence indirect memberships. If the SCIM service provider exposes a "Group" resource, the "value"

sub-attribute MUST be the "id", and the "\$ref" sub-attribute must be the URI of the corresponding "Group" resources to which the user belongs. Since this attribute has a mutability of "readOnly", group membership changes MUST be applied via the "Group" Resource (Section 4.2). This attribute has a mutability of "readOnly".

entitlements

A list of entitlements for the user that represent a thing the user has. An entitlement may be an additional right to a thing, object, or service. No vocabulary or syntax is specified; service providers and clients are expected to encode sufficient information in the value so as to accurately and without ambiguity determine what the user has access to. This value has no canonical types, although a type may be useful as a means to scope entitlements.

roles

A list of roles for the user that collectively represent who the user is, e.g., "Student", "Faculty". No vocabulary or syntax is specified, although it is expected that a role value is a String or label representing a collection of entitlements. This value has no canonical types.

x509Certificates

A list of certificates associated with the resource (e.g., a User). Each value contains exactly one DER-encoded X.509 certificate (see Section 4 of [RFC5280]), which MUST be base64 encoded per Section 4 of [RFC4648]. A single value MUST NOT contain multiple certificates and so does not contain the encoding "SEQUENCE OF Certificate" in any guise.

4.2. "Group" Resource Schema

SCIM provides a schema for representing groups, identified using the following schema URI: "urn:ietf:params:scim:schemas:core:2.0:Group".

"Group" resources are meant to enable expression of common group-based or role-based access control models, although no explicit authorization model is defined. It is intended that the semantics of group membership, and any behavior or authorization granted as a result of membership, are defined by the service provider; these are considered out of scope for this specification.

The following singular attribute is defined in addition to the common attributes defined in the SCIM core schema:

`displayName`

A human-readable name for the Group. REQUIRED.

The following multi-valued attribute is defined in addition to the common attributes defined in the SCIM core schema:

`members`

A list of members of the Group. While values MAY be added or removed, sub-attributes of members are "immutable". The "value" sub-attribute contains the value of an "id" attribute of a SCIM resource, and the "\$ref" sub-attribute must be the URI of a SCIM resource such as a "User", or a "Group". The intention of the "Group" type is to allow the service provider to support nested groups. Service providers MAY require clients to provide a non-empty value by setting the "required" attribute characteristic of a sub-attribute of the "members" attribute in the "Group" resource schema.

4.3. Enterprise User Schema Extension

The following SCIM extension defines attributes commonly used in representing users that belong to, or act on behalf of, a business or enterprise. The enterprise User extension is identified using the following schema URI:

"urn:ietf:params:scim:schemas:extension:enterprise:2.0:User".

The following singular attributes are defined:

`employeeNumber`

A string identifier, typically numeric or alphanumeric, assigned to a person, typically based on order of hire or association with an organization.

`costCenter`

Identifies the name of a cost center.

`organization`

Identifies the name of an organization.

`division`

Identifies the name of a division.

`department`

Identifies the name of a department.

manager

The user's manager. A complex type that optionally allows service providers to represent organizational hierarchy by referencing the "id" attribute of another User.

value The "id" of the SCIM resource representing the user's manager. RECOMMENDED.

\$ref The URI of the SCIM resource representing the User's manager. RECOMMENDED.

displayName The displayName of the user's manager. This attribute is OPTIONAL, and mutability is "readOnly".

5. Service Provider Configuration Schema

SCIM provides a schema for representing the service provider's configuration, identified using the following schema URI: "urn:ietf:params:scim:schemas:core:2.0:ServiceProviderConfig".

The service provider configuration resource enables a service provider to discover SCIM specification features in a standardized form as well as provide additional implementation details to clients. All attributes have a mutability of "readOnly". Unlike other core resources, the "id" attribute is not required for the service provider configuration resource.

The following singular attributes are defined in addition to the common attributes defined in the core schema:

documentationUri

An HTTP-addressable URL pointing to the service provider's human-consumable help documentation. OPTIONAL.

patch

A complex type that specifies PATCH configuration options. REQUIRED. See Section 3.5.2 of [RFC7644].

supported A Boolean value specifying whether or not the operation is supported. REQUIRED.

bulk

A complex type that specifies bulk configuration options. See Section 3.7 of [RFC7644]. REQUIRED.

supported A Boolean value specifying whether or not the operation is supported. REQUIRED.

maxOperations An integer value specifying the maximum number of operations. REQUIRED.

maxPayloadSize An integer value specifying the maximum payload size in bytes. REQUIRED.

filter

A complex type that specifies FILTER options. REQUIRED. See Section 3.4.2.2 of [RFC7644].

supported A Boolean value specifying whether or not the operation is supported. REQUIRED.

maxResults An integer value specifying the maximum number of resources returned in a response. REQUIRED.

changePassword

A complex type that specifies configuration options related to changing a password. REQUIRED.

supported A Boolean value specifying whether or not the operation is supported. REQUIRED.

sort

A complex type that specifies Sort configuration options. REQUIRED.

supported A Boolean value specifying whether or not sorting is supported. REQUIRED.

etag

A complex type that specifies ETag configuration options. REQUIRED.

supported A Boolean value specifying whether or not the operation is supported. REQUIRED.

The following multi-valued attribute is defined in addition to the common attributes defined in the core schema:

authenticationSchemes

A multi-valued complex type that specifies supported authentication scheme properties. To enable seamless discovery of configurations, the service provider SHOULD, with the appropriate security considerations, make the authenticationSchemes attribute publicly accessible without prior authentication. REQUIRED. The following sub-attributes are defined:

type The authentication scheme. This specification defines the values "oauth", "oauth2", "oauthbearer token", "httpbasic", and "httdigest". REQUIRED.

name The common authentication scheme name, e.g., HTTP Basic. REQUIRED.

description A description of the authentication scheme. REQUIRED.

specUri An HTTP-addressable URL pointing to the authentication scheme's specification. OPTIONAL.

documentationUri An HTTP-addressable URL pointing to the authentication scheme's usage documentation. OPTIONAL.

6. ResourceType Schema

The "ResourceType" schema specifies the metadata about a resource type. Resource type resources are READ-ONLY and identified using the following schema URI:

"urn:ietf:params:scim:schemas:core:2.0:ResourceType". Unlike other core resources, all attributes are REQUIRED unless otherwise specified. The "id" attribute is not required for the resource type resource.

The following singular attributes are defined:

id

The resource type's server unique id. This is often the same value as the "name" attribute. OPTIONAL.

name

The resource type name. When applicable, service providers MUST specify the name, e.g., "User" or "Group". This name is referenced by the "meta.resourceType" attribute in all resources. REQUIRED.

description

The resource type's human-readable description. When applicable, service providers MUST specify the description. OPTIONAL.

endpoint

The resource type's HTTP-addressable endpoint relative to the Base URL of the service provider, e.g., "Users". REQUIRED.

schema

The resource type's primary/base schema URI, e.g., "urn:ietf:params:scim:schemas:core:2.0:User". This MUST be equal to the "id" attribute of the associated "Schema" resource. REQUIRED.

schemaExtensions

A list of URIs of the resource type's schema extensions. OPTIONAL.

schema The URI of an extended schema, e.g., "urn:edu:2.0:Staff". This MUST be equal to the "id" attribute of a "Schema" resource. REQUIRED.

required A Boolean value that specifies whether or not the schema extension is required for the resource type. If true, a resource of this type MUST include this schema extension and also include any attributes declared as required in this schema extension. If false, a resource of this type MAY omit this schema extension. REQUIRED.

7. Schema Definition

This section defines a way to specify the schema in use by resources available and accepted by a SCIM service provider. For each "schemas" URI value, this schema specifies the defined attribute(s) and their characteristics (mutability, returnability, etc). For every schema URI used in a resource object, there is a corresponding "Schema" resource. "Schema" resources are not modifiable, and their associated attributes have a mutability of "readOnly". Except for "id" (which is always returned), all attributes have a "returned" characteristic of "default". Unless otherwise specified, all schema attributes are case insensitive. These resources have a "schemas" attribute with the following schema URI:

urn:ietf:params:scim:schemas:core:2.0:Schema

Unlike other core resources, the "Schema" resource MAY contain a complex object within a sub-attribute, and all attributes are REQUIRED unless otherwise specified.

The following singular attributes are defined:

id

The unique URI of the schema. When applicable, service providers MUST specify the URI, e.g., "urn:ietf:params:scim:schemas:core:2.0:User". Unlike most other schemas, which use some sort of Globally Unique Identifier (GUID) for the "id", the schema "id" is a URI so that it can be registered and is portable between different service providers and clients. REQUIRED.

name

The schema's human-readable name. When applicable, service providers MUST specify the name, e.g., "User" or "Group". OPTIONAL.

description

The schema's human-readable description. When applicable, service providers MUST specify the description. OPTIONAL.

The following multi-valued attribute is defined:

attributes

A complex type that defines service provider attributes and their qualities via the following set of sub-attributes:

name The attribute's name.

type The attribute's data type. Valid values are "string", "boolean", "decimal", "integer", "dateTime", "reference", and "complex". When an attribute is of type "complex", there SHOULD be a corresponding schema attribute "subAttributes" defined, listing the sub-attributes of the attribute.

subAttributes When an attribute is of type "complex", "subAttributes" defines a set of sub-attributes. "subAttributes" has the same schema sub-attributes as "attributes".

multiValued A Boolean value indicating the attribute's plurality.

description The attribute's human-readable description. When applicable, service providers MUST specify the description.

required A Boolean value that specifies whether or not the attribute is required.

canonicalValues A collection of suggested canonical values that MAY be used (e.g., "work" and "home"). In some cases, service providers MAY choose to ignore unsupported values. OPTIONAL.

caseExact A Boolean value that specifies whether or not a string attribute is case sensitive. The server SHALL use case sensitivity when evaluating filters. For attributes that are case exact, the server SHALL preserve case for any value submitted. If the attribute is case insensitive, the server MAY alter case for a submitted value. Case sensitivity also impacts how attribute values MAY be compared against filter values (see Section 3.4.2.2 of [RFC7644]).

mutability A single keyword indicating the circumstances under which the value of the attribute can be (re)defined:

readOnly The attribute SHALL NOT be modified.

readWrite The attribute MAY be updated and read at any time. This is the default value.

immutable The attribute MAY be defined at resource creation (e.g., POST) or at record replacement via a request (e.g., a PUT). The attribute SHALL NOT be updated.

writeOnly The attribute MAY be updated at any time. Attribute values SHALL NOT be returned (e.g., because the value is a stored hash). Note: An attribute with a mutability of "writeOnly" usually also has a returned setting of "never".

returned A single keyword that indicates when an attribute and associated values are returned in response to a GET request or in response to a PUT, POST, or PATCH request. Valid keywords are as follows:

always The attribute is always returned, regardless of the contents of the "attributes" parameter. For example, "id" is always returned to identify a SCIM resource.

never The attribute is never returned. This may occur because the original attribute value (e.g., a hashed value) is not retained by the service provider. A service provider MAY allow attributes to be used in a search filter.

default The attribute is returned by default in all SCIM operation responses where attribute values are returned. If the GET request "attributes" parameter is specified, attribute values are only returned if the attribute is named in the "attributes" parameter. DEFAULT.

request The attribute is returned in response to any PUT, POST, or PATCH operations if the attribute was specified by the client (for example, the attribute was modified). The attribute is returned in a SCIM query operation only if specified in the "attributes" parameter.

uniqueness A single keyword value that specifies how the service provider enforces uniqueness of attribute values. A server MAY reject an invalid value based on uniqueness by returning HTTP response code 400 (Bad Request). A client MAY enforce uniqueness on the client side to a greater degree than the service provider enforces. For example, a client could make a value unique while the server has uniqueness of "none". Valid keywords are as follows:

none The values are not intended to be unique in any way. DEFAULT.

server The value SHOULD be unique within the context of the current SCIM endpoint (or tenancy) and MAY be globally unique (e.g., a "username", email address, or other server-generated key or counter). No two resources on the same server SHOULD possess the same value.

global The value SHOULD be globally unique (e.g., an email address, a GUID, or other value). No two resources on any server SHOULD possess the same value.

referenceTypes A multi-valued array of JSON strings that indicate the SCIM resource types that may be referenced. Valid values are as follows:

- + A SCIM resource type (e.g., "User" or "Group"),
- + "external" - indicating that the resource is an external resource (e.g., a photo), or
- + "uri" - indicating that the reference is to a service endpoint or an identifier (e.g., a schema URN).

This attribute is only applicable for attributes that are of type "reference" (Section 2.3.7).

8. JSON Representation

8.1. Minimal User Representation

The following is a non-normative example of the minimal required SCIM representation in JSON format.

```
{
  "schemas": ["urn:ietf:params:scim:schemas:core:2.0:User"],
  "id": "2819c223-7f76-453a-919d-413861904646",
  "userName": "bjensen@example.com",
  "meta": {
    "resourceType": "User",
    "created": "2010-01-23T04:56:22Z",
    "lastModified": "2011-05-13T04:42:34Z",
    "version": "W\/"3694e05e9dff590\"",
    "location":
      "https://example.com/v2/Users/2819c223-7f76-453a-919d-413861904646"
  }
}
```

Figure 3: Example Minimal User JSON Representation

8.2. Full User Representation

The following is a non-normative example of the fully populated SCIM representation in JSON format.

```
{
  "schemas": ["urn:ietf:params:scim:schemas:core:2.0:User"],
  "id": "2819c223-7f76-453a-919d-413861904646",
  "externalId": "701984",
  "userName": "bjensen@example.com",
  "name": {
    "formatted": "Ms. Barbara J Jensen, III",
    "familyName": "Jensen",
    "givenName": "Barbara",
    "middleName": "Jane",
    "honorificPrefix": "Ms.",
    "honorificSuffix": "III"
  },
  "displayName": "Babs Jensen",
  "nickName": "Babs",
  "profileUrl": "https://login.example.com/bjensen",
  "emails": [
    {
      "value": "bjensen@example.com",
      "type": "work",
      "primary": true
    },
    {
      "value": "babs@jensen.org",
      "type": "home"
    }
  ],
  "addresses": [
    {
      "type": "work",
      "streetAddress": "100 Universal City Plaza",
      "locality": "Hollywood",
      "region": "CA",
      "postalCode": "91608",
      "country": "USA",
      "formatted": "100 Universal City Plaza\nHollywood, CA 91608 USA",
      "primary": true
    }
  ],
}
```

```
{
  "type": "home",
  "streetAddress": "456 Hollywood Blvd",
  "locality": "Hollywood",
  "region": "CA",
  "postalCode": "91608",
  "country": "USA",
  "formatted": "456 Hollywood Blvd\nHollywood, CA 91608 USA"
},
"phoneNumbers": [
  {
    "value": "555-555-5555",
    "type": "work"
  },
  {
    "value": "555-555-4444",
    "type": "mobile"
  }
],
"ims": [
  {
    "value": "someaimhandle",
    "type": "aim"
  }
],
"photos": [
  {
    "value":
      "https://photos.example.com/profilephoto/7293000000Ccne/F",
    "type": "photo"
  },
  {
    "value":
      "https://photos.example.com/profilephoto/7293000000Ccne/T",
    "type": "thumbnail"
  }
],
```

```
"userType": "Employee",
"title": "Tour Guide",
"preferredLanguage": "en-US",
"locale": "en-US",
"timezone": "America/Los_Angeles",
"active": true,
"password": "tlmeMa$heen",
"groups": [
  {
    "value": "e9e30dba-f08f-4109-8486-d5c6a331660a",
    "$ref":
"https://example.com/v2/Groups/e9e30dba-f08f-4109-8486-d5c6a331660a",
    "display": "Tour Guides"
  },
  {
    "value": "fc348aa8-3835-40eb-a20b-c726e15c55b5",
    "$ref":
"https://example.com/v2/Groups/fc348aa8-3835-40eb-a20b-c726e15c55b5",
    "display": "Employees"
  },
  {
    "value": "71ddacd2-a8e7-49b8-a5db-ae50d0a5bfd7",
    "$ref":
"https://example.com/v2/Groups/71ddacd2-a8e7-49b8-a5db-ae50d0a5bfd7",
    "display": "US Employees"
  }
],
```

```

"x509Certificates": [
  {
    "value":
      "MIIDQzCCAqygAwIBAgICEAAwDQYJKoZIhvcNAQEFBQAwTjELMAkGA1UEBhMCVVMx
      EzARBgNVBAGMCKNhbGlmb3JuaWExFDASBgNVBAoMC2V4YW1wbGUuY29tMRQwEgYD
      VQDDAtleGFtcGxlLmNvbTAeFw0xMTEwMjI0MzFaFw0xMTEwMjI0MzFaMjEwMzFa
      MH8xCzAJBgNVBAYTAlVTMRMwEQYDVQIDApDYWxpZm9ybmlhMRQwEgYDVQKDA1e
      eGFtcGxlLmNvbTEhMB8GA1UEAwwYTXMuIEJhcmJhcmEgSiBkZW5zZW4gSULJMSIw
      IAYJKoZIhvcNAQkBFhNiamVuc2VuQGv4YW1wbGUuY29tMIIIBIjANBgkqhkiG9w0B
      AQEFAAOCAQ8AMIIBCgKCAQEA7Kr+Dcds/JQ5GwejJFcBIP682X3xpjis56AK02bc
      1FLgzdLI8auoR+cC9/Vrh5t66HkQIOdA4unHh0AaZ4xL5PhVbXIPMB5vAPKpzz5i
      PSi8x08SL7I7SDhcBVJhqVqr3Hg1lEG6UC1DdHO7nkLuwXq8HcISKkbT5WFTVfFZ
      zidP18HZ7DhXkZIRtJwBweq4bvm3hM1Os7UQH05ZS6cVDgweKNwdLLrT51iksQG3
      DYrl+ft781UQRiqxgwqCfXEuDiinPh0kkvIi5jivVulZ9QiwlyEdRbLJ4zJQBmDr
      SGTMYn4lRc2HgHO4DqB/bnMvorHB0CC6AV1QoFK4GPellLwIDAQABo3sweTAJBgNV
      HRMEAjaAMCwGCWCGSAGG+EIBDQQfFh1PcGVuU1NMIEdlbnVvYXRlZCBZDZlZ0aWZp
      Y2F0ZTAdBgNVHQ4EFgQU8pD0U0vsZIsaA16lL8En8bx0F/gwHwYDVR0jBBGwFoAU
      dGeKitcaF7gnzsNwDx708kqaVt0wDQYJKoZIhvcNAQEFBQADgYEEAA81SsFnOdYJt
      Ng5Tcq+/ByEDrBgnusx0jloUhByPMEVkoMZ3J7j1ZgI8rAbOkNngX8+pKfTiDz1R
      C4+dx8oU6Za+4NJXUjllL5CvV6BEYb1+QAEJwitTVvxB/A67g42/vzGAtorUeDovl
      +GFIBZ+GNF/cAYKcMtGcrs2i97ZkJMo="
  }
],
"meta": {
  "resourceType": "User",
  "created": "2010-01-23T04:56:22Z",
  "lastModified": "2011-05-13T04:42:34Z",
  "version": "W\\\\"a330bc54f0671c9\\\"",
  "location":
    "https://example.com/v2/Users/2819c223-7f76-453a-919d-413861904646"
}
}

```

Figure 4: Example Full User JSON Representation

8.3. Enterprise User Extension Representation

The following is a non-normative example of the fully populated User using the enterprise User extension in JSON format.

```
{
  "schemas":
    [ "urn:ietf:params:scim:schemas:core:2.0:User",
      "urn:ietf:params:scim:schemas:extension:enterprise:2.0:User" ],
  "id": "2819c223-7f76-453a-919d-413861904646",
  "externalId": "701984",
  "userName": "bjensen@example.com",
  "name": {
    "formatted": "Ms. Barbara J Jensen, III",
    "familyName": "Jensen",
    "givenName": "Barbara",
    "middleName": "Jane",
    "honorificPrefix": "Ms.",
    "honorificSuffix": "III"
  },
  "displayName": "Babs Jensen",
  "nickName": "Babs",
  "profileUrl": "https://login.example.com/bjensen",
  "emails": [
    {
      "value": "bjensen@example.com",
      "type": "work",
      "primary": true
    },
    {
      "value": "babs@jensen.org",
      "type": "home"
    }
  ],
  "addresses": [
    {
      "streetAddress": "100 Universal City Plaza",
      "locality": "Hollywood",
      "region": "CA",
      "postalCode": "91608",
      "country": "USA",
      "formatted": "100 Universal City Plaza\nHollywood, CA 91608 USA",
      "type": "work",
      "primary": true
    }
  ],
}
```

```
{
  "streetAddress": "456 Hollywood Blvd",
  "locality": "Hollywood",
  "region": "CA",
  "postalCode": "91608",
  "country": "USA",
  "formatted": "456 Hollywood Blvd\nHollywood, CA 91608 USA",
  "type": "home"
},
"phoneNumbers": [
  {
    "value": "555-555-5555",
    "type": "work"
  },
  {
    "value": "555-555-4444",
    "type": "mobile"
  }
],
"ims": [
  {
    "value": "someaimhandle",
    "type": "aim"
  }
],
"photos": [
  {
    "value":
      "https://photos.example.com/profilephoto/7293000000Ccne/F",
    "type": "photo"
  },
  {
    "value":
      "https://photos.example.com/profilephoto/7293000000Ccne/T",
    "type": "thumbnail"
  }
],
```



```
"urn:ietf:params:scim:schemas:extension:enterprise:2.0:User": {
  "employeeNumber": "701984",
  "costCenter": "4130",
  "organization": "Universal Studios",
  "division": "Theme Park",
  "department": "Tour Operations",
  "manager": {
    "value": "26118915-6090-4610-87e4-49d8ca9f808d",
    "$ref": "../Users/26118915-6090-4610-87e4-49d8ca9f808d",
    "displayName": "John Smith"
  }
},
"meta": {
  "resourceType": "User",
  "created": "2010-01-23T04:56:22Z",
  "lastModified": "2011-05-13T04:42:34Z",
  "version": "W\\\\"3694e05e9dff591\\\"",
  "location":
"https://example.com/v2/Users/2819c223-7f76-453a-919d-413861904646"
}
}
```

Figure 5: Example Enterprise User JSON Representation

8.4. Group Representation

The following is a non-normative example of the SCIM Group representation in JSON format.

```
{
  "schemas": [ "urn:ietf:params:scim:schemas:core:2.0:Group" ],
  "id": "e9e30dba-f08f-4109-8486-d5c6a331660a",
  "displayName": "Tour Guides",
  "members": [
    {
      "value": "2819c223-7f76-453a-919d-413861904646",
      "$ref":
        "https://example.com/v2/Users/2819c223-7f76-453a-919d-413861904646",
      "display": "Babs Jensen"
    },
    {
      "value": "902c246b-6245-4190-8e05-00816be7344a",
      "$ref":
        "https://example.com/v2/Users/902c246b-6245-4190-8e05-00816be7344a",
      "display": "Mandy Pepperidge"
    }
  ],
  "meta": {
    "resourceType": "Group",
    "created": "2010-01-23T04:56:22Z",
    "lastModified": "2011-05-13T04:42:34Z",
    "version": "W\/"3694e05e9dff592\"",
    "location":
      "https://example.com/v2/Groups/e9e30dba-f08f-4109-8486-d5c6a331660a"
  }
}
```

Figure 6: Example Group JSON Representation

8.5. Service Provider Configuration Representation

The following is a non-normative example of the SCIM service provider configuration representation in JSON format.

```
{
  "schemas":
    [ "urn:ietf:params:scim:schemas:core:2.0:ServiceProviderConfig" ],
  "documentationUri": "http://example.com/help/scim.html",
  "patch": {
    "supported":true
  },
  "bulk": {
    "supported":true,
    "maxOperations":1000,
    "maxPayloadSize":1048576
  },
  "filter": {
    "supported":true,
    "maxResults": 200
  },
  "changePassword": {
    "supported":true
  },
  "sort": {
    "supported":true
  },
  "etag": {
    "supported":true
  },
  "authenticationSchemes": [
    {
      "name": "OAuth Bearer Token",
      "description":
        "Authentication scheme using the OAuth Bearer Token Standard",
      "specUri": "http://www.rfc-editor.org/info/rfc6750",
      "documentationUri": "http://example.com/help/oauth.html",
      "type": "oauthbearertoken",
      "primary": true
    }
  ],
}
```

```
{
  "name": "HTTP Basic",
  "description":
    "Authentication scheme using the HTTP Basic Standard",
  "specUri": "http://www.rfc-editor.org/info/rfc2617",
  "documentationUri": "http://example.com/help/httpBasic.html",
  "type": "httpbasic"
},
"meta": {
  "location": "https://example.com/v2/ServiceProviderConfig",
  "resourceType": "ServiceProviderConfig",
  "created": "2010-01-23T04:56:22Z",
  "lastModified": "2011-05-13T04:42:34Z",
  "version": "W\/"3694e05e9dff594\""}
}
```

Figure 7: Example Service Provider Configuration JSON Representation

8.6. Resource Type Representation

The following is a non-normative example of the SCIM resource types in JSON format.

```
[{
  "schemas": ["urn:ietf:params:scim:schemas:core:2.0:ResourceType"],
  "id": "User",
  "name": "User",
  "endpoint": "/Users",
  "description": "User Account",
  "schema": "urn:ietf:params:scim:schemas:core:2.0:User",
  "schemaExtensions": [
    {
      "schema":
        "urn:ietf:params:scim:schemas:extension:enterprise:2.0:User",
      "required": true
    }
  ],
  "meta": {
    "location": "https://example.com/v2/ResourceTypes/User",
    "resourceType": "ResourceType"
  }
},
{
  "schemas": ["urn:ietf:params:scim:schemas:core:2.0:ResourceType"],
  "id": "Group",
  "name": "Group",
  "endpoint": "/Groups",
  "description": "Group",
  "schema": "urn:ietf:params:scim:schemas:core:2.0:Group",
  "meta": {
    "location": "https://example.com/v2/ResourceTypes/Group",
    "resourceType": "ResourceType"
  }
}
]
```

Figure 8: Example Resource Type JSON Representation

8.7. Schema Representation

The following sections provide representations of schemas for both SCIM resources and service provider schemas. Note that the JSON representation has been modified for readability and to fit the specification format.

8.7.1. Resource Schema Representation

The following is intended as an example of the SCIM schema representation in JSON format for SCIM resources. Where permitted, individual values and schema MAY change. This example includes schema representations for "User", "Group", and "EnterpriseUser"; other schema representations are possible.

```
[
  {
    "id" : "urn:ietf:params:scim:schemas:core:2.0:User",
    "name" : "User",
    "description" : "User Account",
    "attributes" : [
      {
        "name" : "userName",
        "type" : "string",
        "multiValued" : false,
        "description" : "Unique identifier for the User, typically
used by the user to directly authenticate to the service provider.
Each User MUST include a non-empty userName value. This identifier
MUST be unique across the service provider's entire set of Users.
REQUIRED.",
        "required" : true,
        "caseExact" : false,
        "mutability" : "readWrite",
        "returned" : "default",
        "uniqueness" : "server"
      },
    ],
  },
]
```

```

    {
      "name" : "name",
      "type" : "complex",
      "multiValued" : false,
      "description" : "The components of the user's real name.
Providers MAY return just the full name as a single string in the
formatted sub-attribute, or they MAY return just the individual
component attributes using the other sub-attributes, or they MAY
return both. If both variants are returned, they SHOULD be
describing the same name, with the formatted name indicating how the
component attributes should be combined.",
      "required" : false,
      "subAttributes" : [
        {
          "name" : "formatted",
          "type" : "string",
          "multiValued" : false,
          "description" : "The full name, including all middle
names, titles, and suffixes as appropriate, formatted for display
(e.g., 'Ms. Barbara J Jensen, III').",
          "required" : false,
          "caseExact" : false,
          "mutability" : "readWrite",
          "returned" : "default",
          "uniqueness" : "none"
        },
        {
          "name" : "familyName",
          "type" : "string",
          "multiValued" : false,
          "description" : "The family name of the User, or
last name in most Western languages (e.g., 'Jensen' given the full
name 'Ms. Barbara J Jensen, III').",
          "required" : false,
          "caseExact" : false,
          "mutability" : "readWrite",
          "returned" : "default",
          "uniqueness" : "none"
        }
      ],
    },
  ],

```

```
{
  "name" : "givenName",
  "type" : "string",
  "multiValued" : false,
  "description" : "The given name of the User, or
first name in most Western languages (e.g., 'Barbara' given the
full name 'Ms. Barbara J Jensen, III').",
  "required" : false,
  "caseExact" : false,
  "mutability" : "readWrite",
  "returned" : "default",
  "uniqueness" : "none"
},
{
  "name" : "middleName",
  "type" : "string",
  "multiValued" : false,
  "description" : "The middle name(s) of the User
(e.g., 'Jane' given the full name 'Ms. Barbara J Jensen, III').",
  "required" : false,
  "caseExact" : false,
  "mutability" : "readWrite",
  "returned" : "default",
  "uniqueness" : "none"
},
{
  "name" : "honorificPrefix",
  "type" : "string",
  "multiValued" : false,
  "description" : "The honorific prefix(es) of the User, or
title in most Western languages (e.g., 'Ms.' given the full name
'Ms. Barbara J Jensen, III').",
  "required" : false,
  "caseExact" : false,
  "mutability" : "readWrite",
  "returned" : "default",
  "uniqueness" : "none"
},
}
```

```
    {
      "name" : "honorificSuffix",
      "type" : "string",
      "multiValued" : false,
      "description" : "The honorific suffix(es) of the User, or
suffix in most Western languages (e.g., 'III' given the full name
'Ms. Barbara J Jensen, III').",
      "required" : false,
      "caseExact" : false,
      "mutability" : "readWrite",
      "returned" : "default",
      "uniqueness" : "none"
    }
  ],
  "mutability" : "readWrite",
  "returned" : "default",
  "uniqueness" : "none"
},
{
  "name" : "displayName",
  "type" : "string",
  "multiValued" : false,
  "description" : "The name of the User, suitable for display
to end-users. The name SHOULD be the full name of the User being
described, if known.",
  "required" : false,
  "caseExact" : false,
  "mutability" : "readWrite",
  "returned" : "default",
  "uniqueness" : "none"
},
{
  "name" : "nickName",
  "type" : "string",
  "multiValued" : false,
  "description" : "The casual way to address the user in real
life, e.g., 'Bob' or 'Bobby' instead of 'Robert'. This attribute
SHOULD NOT be used to represent a User's username (e.g., 'bjensen' or
'mpepperidge').",
  "required" : false,
  "caseExact" : false,
  "mutability" : "readWrite",
  "returned" : "default",
  "uniqueness" : "none"
},
},
```

```
{
  "name" : "profileUrl",
  "type" : "reference",
  "referenceTypes" : ["external"],
  "multiValued" : false,
  "description" : "A fully qualified URL pointing to a page
representing the User's online profile.",
  "required" : false,
  "caseExact" : false,
  "mutability" : "readWrite",
  "returned" : "default",
  "uniqueness" : "none"
},
{
  "name" : "title",
  "type" : "string",
  "multiValued" : false,
  "description" : "The user's title, such as
\"Vice President.\",
  "required" : false,
  "caseExact" : false,
  "mutability" : "readWrite",
  "returned" : "default",
  "uniqueness" : "none"
},
{
  "name" : "userType",
  "type" : "string",
  "multiValued" : false,
  "description" : "Used to identify the relationship between
the organization and the user. Typical values used might be
'Contractor', 'Employee', 'Intern', 'Temp', 'External', and
'Unknown', but any value may be used.",
  "required" : false,
  "caseExact" : false,
  "mutability" : "readWrite",
  "returned" : "default",
  "uniqueness" : "none"
},
},
```

```
{
  "name" : "preferredLanguage",
  "type" : "string",
  "multiValued" : false,
  "description" : "Indicates the User's preferred written or
spoken language. Generally used for selecting a localized user
interface; e.g., 'en_US' specifies the language English and country
US.",
  "required" : false,
  "caseExact" : false,
  "mutability" : "readWrite",
  "returned" : "default",
  "uniqueness" : "none"
},
{
  "name" : "locale",
  "type" : "string",
  "multiValued" : false,
  "description" : "Used to indicate the User's default location
for purposes of localizing items such as currency, date time format, or
numerical representations.",
  "required" : false,
  "caseExact" : false,
  "mutability" : "readWrite",
  "returned" : "default",
  "uniqueness" : "none"
},
{
  "name" : "timezone",
  "type" : "string",
  "multiValued" : false,
  "description" : "The User's time zone in the 'Olson' time zone
database format, e.g., 'America/Los_Angeles'.",
  "required" : false,
  "caseExact" : false,
  "mutability" : "readWrite",
  "returned" : "default",
  "uniqueness" : "none"
},
}
```

```

    {
      "name" : "active",
      "type" : "boolean",
      "multiValued" : false,
      "description" : "A Boolean value indicating the User's
administrative status.",
      "required" : false,
      "mutability" : "readWrite",
      "returned" : "default"
    },
    {
      "name" : "password",
      "type" : "string",
      "multiValued" : false,
      "description" : "The User's cleartext password. This
attribute is intended to be used as a means to specify an initial
password when creating a new User or to reset an existing User's
password.",
      "required" : false,
      "caseExact" : false,
      "mutability" : "writeOnly",
      "returned" : "never",
      "uniqueness" : "none"
    },
    {
      "name" : "emails",
      "type" : "complex",
      "multiValued" : true,
      "description" : "Email addresses for the user. The value
SHOULD be canonicalized by the service provider, e.g.,
'bjensen@example.com' instead of 'bjensen@EXAMPLE.COM'.
Canonical type values of 'work', 'home', and 'other'.",
      "required" : false,
      "subAttributes" : [
        {
          "name" : "value",
          "type" : "string",
          "multiValued" : false,
          "description" : "Email addresses for the user. The value
SHOULD be canonicalized by the service provider, e.g.,
'bjensen@example.com' instead of 'bjensen@EXAMPLE.COM'.
Canonical type values of 'work', 'home', and 'other'.",
          "required" : false,
          "caseExact" : false,
          "mutability" : "readWrite",
          "returned" : "default",
          "uniqueness" : "none"
        }
      ]
    },
  ],

```

```

    {
      "name" : "display",
      "type" : "string",
      "multiValued" : false,
      "description" : "A human-readable name, primarily used
for display purposes. READ-ONLY.",
      "required" : false,
      "caseExact" : false,
      "mutability" : "readWrite",
      "returned" : "default",
      "uniqueness" : "none"
    },
    {
      "name" : "type",
      "type" : "string",
      "multiValued" : false,
      "description" : "A label indicating the attribute's
function, e.g., 'work' or 'home'.",
      "required" : false,
      "caseExact" : false,
      "canonicalValues" : [
        "work",
        "home",
        "other"
      ],
      "mutability" : "readWrite",
      "returned" : "default",
      "uniqueness" : "none"
    },
    {
      "name" : "primary",
      "type" : "boolean",
      "multiValued" : false,
      "description" : "A Boolean value indicating the 'primary'
or preferred attribute value for this attribute, e.g., the preferred
mailing address or primary email address. The primary attribute
value 'true' MUST appear no more than once.",
      "required" : false,
      "mutability" : "readWrite",
      "returned" : "default"
    }
  ],
  "mutability" : "readWrite",
  "returned" : "default",
  "uniqueness" : "none"
},

```

```
{
  "name" : "phoneNumbers",
  "type" : "complex",
  "multiValued" : true,
  "description" : "Phone numbers for the User. The value
SHOULD be canonicalized by the service provider according to the
format specified in RFC 3966, e.g., 'tel:+1-201-555-0123'.
Canonical type values of 'work', 'home', 'mobile', 'fax', 'pager',
and 'other'." ,
  "required" : false,
  "subAttributes" : [
    {
      "name" : "value",
      "type" : "string",
      "multiValued" : false,
      "description" : "Phone number of the User.",
      "required" : false,
      "caseExact" : false,
      "mutability" : "readWrite",
      "returned" : "default",
      "uniqueness" : "none"
    },
    {
      "name" : "display",
      "type" : "string",
      "multiValued" : false,
      "description" : "A human-readable name, primarily used
for display purposes. READ-ONLY.",
      "required" : false,
      "caseExact" : false,
      "mutability" : "readWrite",
      "returned" : "default",
      "uniqueness" : "none"
    }
  ],
}
```

```
    {
      "name" : "type",
      "type" : "string",
      "multiValued" : false,
      "description" : "A label indicating the attribute's
function, e.g., 'work', 'home', 'mobile'.",
      "required" : false,
      "caseExact" : false,
      "canonicalValues" : [
        "work",
        "home",
        "mobile",
        "fax",
        "pager",
        "other"
      ],
      "mutability" : "readWrite",
      "returned" : "default",
      "uniqueness" : "none"
    },
    {
      "name" : "primary",
      "type" : "boolean",
      "multiValued" : false,
      "description" : "A Boolean value indicating the 'primary'
or preferred attribute value for this attribute, e.g., the preferred
phone number or primary phone number. The primary attribute value
'true' MUST appear no more than once.",
      "required" : false,
      "mutability" : "readWrite",
      "returned" : "default"
    }
  ],
  "mutability" : "readWrite",
  "returned" : "default"
},
```

```
{
  "name" : "ims",
  "type" : "complex",
  "multiValued" : true,
  "description" : "Instant messaging addresses for the User.",
  "required" : false,
  "subAttributes" : [
    {
      "name" : "value",
      "type" : "string",
      "multiValued" : false,
      "description" : "Instant messaging address for the User.",
      "required" : false,
      "caseExact" : false,
      "mutability" : "readWrite",
      "returned" : "default",
      "uniqueness" : "none"
    },
    {
      "name" : "display",
      "type" : "string",
      "multiValued" : false,
      "description" : "A human-readable name, primarily used
for display purposes. READ-ONLY.",
      "required" : false,
      "caseExact" : false,
      "mutability" : "readWrite",
      "returned" : "default",
      "uniqueness" : "none"
    }
  ],
}
```

```

    {
      "name" : "type",
      "type" : "string",
      "multiValued" : false,
      "description" : "A label indicating the attribute's
function, e.g., 'aim', 'gtalk', 'xmpp'.",
      "required" : false,
      "caseExact" : false,
      "canonicalValues" : [
        "aim",
        "gtalk",
        "icq",
        "xmpp",
        "msn",
        "skype",
        "qq",
        "yahoo"
      ],
      "mutability" : "readWrite",
      "returned" : "default",
      "uniqueness" : "none"
    },
    {
      "name" : "primary",
      "type" : "boolean",
      "multiValued" : false,
      "description" : "A Boolean value indicating the 'primary'
or preferred attribute value for this attribute, e.g., the preferred
messenger or primary messenger. The primary attribute value 'true'
MUST appear no more than once.",
      "required" : false,
      "mutability" : "readWrite",
      "returned" : "default"
    }
  ],
  "mutability" : "readWrite",
  "returned" : "default"
},

```

```
{
  "name" : "photos",
  "type" : "complex",
  "multiValued" : true,
  "description" : "URLs of photos of the User.",
  "required" : false,
  "subAttributes" : [
    {
      "name" : "value",
      "type" : "reference",
      "referenceTypes" : ["external"],
      "multiValued" : false,
      "description" : "URL of a photo of the User.",
      "required" : false,
      "caseExact" : false,
      "mutability" : "readWrite",
      "returned" : "default",
      "uniqueness" : "none"
    },
    {
      "name" : "display",
      "type" : "string",
      "multiValued" : false,
      "description" : "A human-readable name, primarily used
for display purposes. READ-ONLY.",
      "required" : false,
      "caseExact" : false,
      "mutability" : "readWrite",
      "returned" : "default",
      "uniqueness" : "none"
    }
  ],
}
```

```
    {
      "name" : "type",
      "type" : "string",
      "multiValued" : false,
      "description" : "A label indicating the attribute's
function, i.e., 'photo' or 'thumbnail'." ,
      "required" : false,
      "caseExact" : false,
      "canonicalValues" : [
        "photo",
        "thumbnail"
      ],
      "mutability" : "readWrite",
      "returned" : "default",
      "uniqueness" : "none"
    },
    {
      "name" : "primary",
      "type" : "boolean",
      "multiValued" : false,
      "description" : "A Boolean value indicating the 'primary'
or preferred attribute value for this attribute, e.g., the preferred
photo or thumbnail. The primary attribute value 'true' MUST appear
no more than once.",
      "required" : false,
      "mutability" : "readWrite",
      "returned" : "default"
    }
  ],
  "mutability" : "readWrite",
  "returned" : "default"
},
```

```

    {
      "name" : "addresses",
      "type" : "complex",
      "multiValued" : true,
      "description" : "A physical mailing address for this User.
Canonical type values of 'work', 'home', and 'other'. This attribute
is a complex type with the following sub-attributes.",
      "required" : false,
      "subAttributes" : [
        {
          "name" : "formatted",
          "type" : "string",
          "multiValued" : false,
          "description" : "The full mailing address, formatted for
display or use with a mailing label. This attribute MAY contain
newlines.",
          "required" : false,
          "caseExact" : false,
          "mutability" : "readWrite",
          "returned" : "default",
          "uniqueness" : "none"
        },
        {
          "name" : "streetAddress",
          "type" : "string",
          "multiValued" : false,
          "description" : "The full street address component,
which may include house number, street name, P.O. box, and multi-line
extended street address information. This attribute MAY contain
newlines.",
          "required" : false,
          "caseExact" : false,
          "mutability" : "readWrite",
          "returned" : "default",
          "uniqueness" : "none"
        },
        {
          "name" : "locality",
          "type" : "string",
          "multiValued" : false,
          "description" : "The city or locality component.",
          "required" : false,
          "caseExact" : false,
          "mutability" : "readWrite",
          "returned" : "default",
          "uniqueness" : "none"
        }
      ]
    },

```

```
{
  "name" : "region",
  "type" : "string",
  "multiValued" : false,
  "description" : "The state or region component.",
  "required" : false,
  "caseExact" : false,
  "mutability" : "readWrite",
  "returned" : "default",
  "uniqueness" : "none"
},
{
  "name" : "postalCode",
  "type" : "string",
  "multiValued" : false,
  "description" : "The zip code or postal code component.",
  "required" : false,
  "caseExact" : false,
  "mutability" : "readWrite",
  "returned" : "default",
  "uniqueness" : "none"
},
{
  "name" : "country",
  "type" : "string",
  "multiValued" : false,
  "description" : "The country name component.",
  "required" : false,
  "caseExact" : false,
  "mutability" : "readWrite",
  "returned" : "default",
  "uniqueness" : "none"
},
},
```

```

    {
      "name" : "type",
      "type" : "string",
      "multiValued" : false,
      "description" : "A label indicating the attribute's
function, e.g., 'work' or 'home'.",
      "required" : false,
      "caseExact" : false,
      "canonicalValues" : [
        "work",
        "home",
        "other"
      ],
      "mutability" : "readWrite",
      "returned" : "default",
      "uniqueness" : "none"
    }
  ],
  "mutability" : "readWrite",
  "returned" : "default",
  "uniqueness" : "none"
},
{
  "name" : "groups",
  "type" : "complex",
  "multiValued" : true,
  "description" : "A list of groups to which the user belongs,
either through direct membership, through nested groups, or
dynamically calculated.",
  "required" : false,
  "subAttributes" : [
    {
      "name" : "value",
      "type" : "string",
      "multiValued" : false,
      "description" : "The identifier of the User's group.",
      "required" : false,
      "caseExact" : false,
      "mutability" : "readOnly",
      "returned" : "default",
      "uniqueness" : "none"
    }
  ],

```

```

    {
      "name" : "$ref",
      "type" : "reference",
      "referenceTypes" : [
        "User",
        "Group"
      ],
      "multiValued" : false,
      "description" : "The URI of the corresponding 'Group'
resource to which the user belongs.",
      "required" : false,
      "caseExact" : false,
      "mutability" : "readOnly",
      "returned" : "default",
      "uniqueness" : "none"
    },
    {
      "name" : "display",
      "type" : "string",
      "multiValued" : false,
      "description" : "A human-readable name, primarily used
for display purposes. READ-ONLY.",
      "required" : false,
      "caseExact" : false,
      "mutability" : "readOnly",
      "returned" : "default",
      "uniqueness" : "none"
    },
    {
      "name" : "type",
      "type" : "string",
      "multiValued" : false,
      "description" : "A label indicating the attribute's
function, e.g., 'direct' or 'indirect'.",
      "required" : false,
      "caseExact" : false,
      "canonicalValues" : [
        "direct",
        "indirect"
      ],
      "mutability" : "readOnly",
      "returned" : "default",
      "uniqueness" : "none"
    }
  ],
  "mutability" : "readOnly",
  "returned" : "default"
},

```

```
{
  "name" : "entitlements",
  "type" : "complex",
  "multiValued" : true,
  "description" : "A list of entitlements for the User that
represent a thing the User has.",
  "required" : false,
  "subAttributes" : [
    {
      "name" : "value",
      "type" : "string",
      "multiValued" : false,
      "description" : "The value of an entitlement.",
      "required" : false,
      "caseExact" : false,
      "mutability" : "readWrite",
      "returned" : "default",
      "uniqueness" : "none"
    },
    {
      "name" : "display",
      "type" : "string",
      "multiValued" : false,
      "description" : "A human-readable name, primarily used
for display purposes. READ-ONLY.",
      "required" : false,
      "caseExact" : false,
      "mutability" : "readWrite",
      "returned" : "default",
      "uniqueness" : "none"
    },
    {
      "name" : "type",
      "type" : "string",
      "multiValued" : false,
      "description" : "A label indicating the attribute's
function.",
      "required" : false,
      "caseExact" : false,
      "mutability" : "readWrite",
      "returned" : "default",
      "uniqueness" : "none"
    }
  ],
}
```

```

    {
      "name" : "primary",
      "type" : "boolean",
      "multiValued" : false,
      "description" : "A Boolean value indicating the 'primary'
or preferred attribute value for this attribute. The primary
attribute value 'true' MUST appear no more than once.",
      "required" : false,
      "mutability" : "readWrite",
      "returned" : "default"
    }
  ],
  "mutability" : "readWrite",
  "returned" : "default"
},
{
  "name" : "roles",
  "type" : "complex",
  "multiValued" : true,
  "description" : "A list of roles for the User that
collectively represent who the User is, e.g., 'Student', 'Faculty'.",
  "required" : false,
  "subAttributes" : [
    {
      "name" : "value",
      "type" : "string",
      "multiValued" : false,
      "description" : "The value of a role.",
      "required" : false,
      "caseExact" : false,
      "mutability" : "readWrite",
      "returned" : "default",
      "uniqueness" : "none"
    },
    {
      "name" : "display",
      "type" : "string",
      "multiValued" : false,
      "description" : "A human-readable name, primarily used
for display purposes. READ-ONLY.",
      "required" : false,
      "caseExact" : false,
      "mutability" : "readWrite",
      "returned" : "default",
      "uniqueness" : "none"
    }
  ],

```

```

    {
      "name" : "type",
      "type" : "string",
      "multiValued" : false,
      "description" : "A label indicating the attribute's
function.",
      "required" : false,
      "caseExact" : false,
      "canonicalValues" : [],
      "mutability" : "readWrite",
      "returned" : "default",
      "uniqueness" : "none"
    },
    {
      "name" : "primary",
      "type" : "boolean",
      "multiValued" : false,
      "description" : "A Boolean value indicating the 'primary'
or preferred attribute value for this attribute. The primary
attribute value 'true' MUST appear no more than once.",
      "required" : false,
      "mutability" : "readWrite",
      "returned" : "default"
    }
  ],
  "mutability" : "readWrite",
  "returned" : "default"
},
{
  "name" : "x509Certificates",
  "type" : "complex",
  "multiValued" : true,
  "description" : "A list of certificates issued to the User.",
  "required" : false,
  "caseExact" : false,
  "subAttributes" : [
    {
      "name" : "value",
      "type" : "binary",
      "multiValued" : false,
      "description" : "The value of an X.509 certificate.",
      "required" : false,
      "caseExact" : false,
      "mutability" : "readWrite",
      "returned" : "default",
      "uniqueness" : "none"
    }
  ],

```

```

    {
      "name" : "display",
      "type" : "string",
      "multiValued" : false,
      "description" : "A human-readable name, primarily used
for display purposes. READ-ONLY.",
      "required" : false,
      "caseExact" : false,
      "mutability" : "readWrite",
      "returned" : "default",
      "uniqueness" : "none"
    },
    {
      "name" : "type",
      "type" : "string",
      "multiValued" : false,
      "description" : "A label indicating the attribute's
function.",
      "required" : false,
      "caseExact" : false,
      "canonicalValues" : [],
      "mutability" : "readWrite",
      "returned" : "default",
      "uniqueness" : "none"
    },
    {
      "name" : "primary",
      "type" : "boolean",
      "multiValued" : false,
      "description" : "A Boolean value indicating the 'primary'
or preferred attribute value for this attribute. The primary
attribute value 'true' MUST appear no more than once.",
      "required" : false,
      "mutability" : "readWrite",
      "returned" : "default"
    }
  ],
  "mutability" : "readWrite",
  "returned" : "default"
}
],
"meta" : {
  "resourceType" : "Schema",
  "location" :
    "/v2/Schemas/urn:ietf:params:scim:schemas:core:2.0:User"
}
},

```

```
{
  "id" : "urn:ietf:params:scim:schemas:core:2.0:Group",
  "name" : "Group",
  "description" : "Group",
  "attributes" : [
    {
      "name" : "displayName",
      "type" : "string",
      "multiValued" : false,
      "description" : "A human-readable name for the Group.
REQUIRED.",
      "required" : false,
      "caseExact" : false,
      "mutability" : "readWrite",
      "returned" : "default",
      "uniqueness" : "none"
    },
    {
      "name" : "members",
      "type" : "complex",
      "multiValued" : true,
      "description" : "A list of members of the Group.",
      "required" : false,
      "subAttributes" : [
        {
          "name" : "value",
          "type" : "string",
          "multiValued" : false,
          "description" : "Identifier of the member of this Group.",
          "required" : false,
          "caseExact" : false,
          "mutability" : "immutable",
          "returned" : "default",
          "uniqueness" : "none"
        }
      ]
    }
  ],
}
```

```

    {
      "name" : "$ref",
      "type" : "reference",
      "referenceTypes" : [
        "User",
        "Group"
      ],
      "multiValued" : false,
      "description" : "The URI corresponding to a SCIM resource
that is a member of this Group.",
      "required" : false,
      "caseExact" : false,
      "mutability" : "immutable",
      "returned" : "default",
      "uniqueness" : "none"
    },
    {
      "name" : "type",
      "type" : "string",
      "multiValued" : false,
      "description" : "A label indicating the type of resource,
e.g., 'User' or 'Group'.",
      "required" : false,
      "caseExact" : false,
      "canonicalValues" : [
        "User",
        "Group"
      ],
      "mutability" : "immutable",
      "returned" : "default",
      "uniqueness" : "none"
    }
  ],
  "mutability" : "readWrite",
  "returned" : "default"
}
],
"meta" : {
  "resourceType" : "Schema",
  "location" :
    "/v2/Schemas/urn:ietf:params:scim:schemas:core:2.0:Group"
}
},

```

```
{
  "id" : "urn:ietf:params:scim:schemas:extension:enterprise:2.0:User",
  "name" : "EnterpriseUser",
  "description" : "Enterprise User",
  "attributes" : [
    {
      "name" : "employeeNumber",
      "type" : "string",
      "multiValued" : false,
      "description" : "Numeric or alphanumeric identifier assigned
to a person, typically based on order of hire or association with an
organization.",
      "required" : false,
      "caseExact" : false,
      "mutability" : "readWrite",
      "returned" : "default",
      "uniqueness" : "none"
    },
    {
      "name" : "costCenter",
      "type" : "string",
      "multiValued" : false,
      "description" : "Identifies the name of a cost center.",
      "required" : false,
      "caseExact" : false,
      "mutability" : "readWrite",
      "returned" : "default",
      "uniqueness" : "none"
    },
    {
      "name" : "organization",
      "type" : "string",
      "multiValued" : false,
      "description" : "Identifies the name of an organization.",
      "required" : false,
      "caseExact" : false,
      "mutability" : "readWrite",
      "returned" : "default",
      "uniqueness" : "none"
    }
  ],
}
```

```
{
  "name" : "division",
  "type" : "string",
  "multiValued" : false,
  "description" : "Identifies the name of a division.",
  "required" : false,
  "caseExact" : false,
  "mutability" : "readWrite",
  "returned" : "default",
  "uniqueness" : "none"
},
{
  "name" : "department",
  "type" : "string",
  "multiValued" : false,
  "description" : "Identifies the name of a department.",
  "required" : false,
  "caseExact" : false,
  "mutability" : "readWrite",
  "returned" : "default",
  "uniqueness" : "none"
},
{
  "name" : "manager",
  "type" : "complex",
  "multiValued" : false,
  "description" : "The User's manager. A complex type that
optionally allows service providers to represent organizational
hierarchy by referencing the 'id' attribute of another User.",
  "required" : false,
  "subAttributes" : [
    {
      "name" : "value",
      "type" : "string",
      "multiValued" : false,
      "description" : "The id of the SCIM resource representing
the User's manager. REQUIRED.",
      "required" : false,
      "caseExact" : false,
      "mutability" : "readWrite",
      "returned" : "default",
      "uniqueness" : "none"
    }
  ]
},
```

```

    {
      "name" : "$ref",
      "type" : "reference",
      "referenceTypes" : [
        "User"
      ],
      "multiValued" : false,
      "description" : "The URI of the SCIM resource
representing the User's manager.  REQUIRED.",
      "required" : false,
      "caseExact" : false,
      "mutability" : "readWrite",
      "returned" : "default",
      "uniqueness" : "none"
    },
    {
      "name" : "displayName",
      "type" : "string",
      "multiValued" : false,
      "description" : "The displayName of the User's manager.
OPTIONAL and READ-ONLY.",
      "required" : false,
      "caseExact" : false,
      "mutability" : "readOnly",
      "returned" : "default",
      "uniqueness" : "none"
    }
  ],
  "mutability" : "readWrite",
  "returned" : "default"
}
],
"meta" : {
  "resourceType" : "Schema",
  "location" :
"/v2/Schemas/urn:ietf:params:scim:schemas:extension:enterprise:2.0:User"
}
}
]

```

Figure 9: Example JSON Representation for Resource Schema

8.7.2. Service Provider Schema Representation

The following is a representation of the SCIM schema for the fixed service provider schemas: ServiceProviderConfig, ResourceType, and Schema.

```
[
  {
    "id" :
      "urn:ietf:params:scim:schemas:core:2.0:ServiceProviderConfig",
    "name" : "Service Provider Configuration",
    "description" : "Schema for representing the service provider's
      configuration",
    "attributes" : [
      {
        "name" : "documentationUri",
        "type" : "reference",
        "referenceTypes" : ["external"],
        "multiValued" : false,
        "description" : "An HTTP-addressable URL pointing to the
          service provider's human-consumable help documentation.",
        "required" : false,
        "caseExact" : false,
        "mutability" : "readOnly",
        "returned" : "default",
        "uniqueness" : "none"
      },
    ],
  },
]
```

```
{
  "name" : "patch",
  "type" : "complex",
  "multiValued" : false,
  "description" : "A complex type that specifies PATCH
    configuration options.",
  "required" : true,
  "returned" : "default",
  "mutability" : "readOnly",
  "subAttributes" : [
    {
      "name" : "supported",
      "type" : "boolean",
      "multiValued" : false,
      "description" : "A Boolean value specifying whether or not
        the operation is supported.",
      "required" : true,
      "mutability" : "readOnly",
      "returned" : "default"
    }
  ]
},
{
  "name" : "bulk",
  "type" : "complex",
  "multiValued" : false,
  "description" : "A complex type that specifies bulk
    configurations.",
  "required" : true,
  "returned" : "default",
  "mutability" : "readOnly",
  "subAttributes" : [
    {
      "name" : "supported",
      "type" : "boolean",
      "multiValued" : false,
      "description" : "A Boolean value specifying whether or not
        the operation is supported.",
      "required" : true,
      "mutability" : "readOnly",
      "returned" : "default"
    }
  ]
},
```

```
{
  "name" : "maxOperations",
  "type" : "integer",
  "multiValued" : false,
  "description" : "An integer value specifying the maximum
    number of operations.",
  "required" : true,
  "mutability" : "readOnly",
  "returned" : "default",
  "uniqueness" : "none"
},
{
  "name" : "maxPayloadSize",
  "type" : "integer",
  "multiValued" : false,
  "description" : "An integer value specifying the maximum
    payload size in bytes.",
  "required" : true,
  "mutability" : "readOnly",
  "returned" : "default",
  "uniqueness" : "none"
}
]
},
{
  "name" : "filter",
  "type" : "complex",
  "multiValued" : false,
  "description" : "A complex type that specifies
    FILTER options.",
  "required" : true,
  "returned" : "default",
  "mutability" : "readOnly",
  "subAttributes" : [
    {
      "name" : "supported",
      "type" : "boolean",
      "multiValued" : false,
      "description" : "A Boolean value specifying whether or not
        the operation is supported.",
      "required" : true,
      "mutability" : "readOnly",
      "returned" : "default"
    }
  ],
}
```

```
{
  "name" : "maxResults",
  "type" : "integer",
  "multiValued" : false,
  "description" : "An integer value specifying the maximum
    number of resources returned in a response.",
  "required" : true,
  "mutability" : "readOnly",
  "returned" : "default",
  "uniqueness" : "none"
}
],
{
  "name" : "changePassword",
  "type" : "complex",
  "multiValued" : false,
  "description" : "A complex type that specifies configuration
    options related to changing a password.",
  "required" : true,
  "returned" : "default",
  "mutability" : "readOnly",
  "subAttributes" : [
    {
      "name" : "supported",
      "type" : "boolean",
      "multiValued" : false,
      "description" : "A Boolean value specifying whether or not
        the operation is supported.",
      "required" : true,
      "mutability" : "readOnly",
      "returned" : "default"
    }
  ]
},
```

```
{
  "name" : "sort",
  "type" : "complex",
  "multiValued" : false,
  "description" : "A complex type that specifies sort result
    options.",
  "required" : true,
  "returned" : "default",
  "mutability" : "readOnly",
  "subAttributes" : [
    {
      "name" : "supported",
      "type" : "boolean",
      "multiValued" : false,
      "description" : "A Boolean value specifying whether or not
        the operation is supported.",
      "required" : true,
      "mutability" : "readOnly",
      "returned" : "default"
    }
  ]
},
{
  "name" : "authenticationSchemes",
  "type" : "complex",
  "multiValued" : true,
  "description" : "A complex type that specifies supported
    authentication scheme properties.",
  "required" : true,
  "returned" : "default",
  "mutability" : "readOnly",
  "subAttributes" : [
    {
      "name" : "name",
      "type" : "string",
      "multiValued" : false,
      "description" : "The common authentication scheme name,
        e.g., HTTP Basic.",
      "required" : true,
      "caseExact" : false,
      "mutability" : "readOnly",
      "returned" : "default",
      "uniqueness" : "none"
    }
  ],
}
```

```
{
  "name" : "description",
  "type" : "string",
  "multiValued" : false,
  "description" : "A description of the authentication
    scheme.",
  "required" : true,
  "caseExact" : false,
  "mutability" : "readOnly",
  "returned" : "default",
  "uniqueness" : "none"
},
{
  "name" : "specUri",
  "type" : "reference",
  "referenceTypes" : ["external"],
  "multiValued" : false,
  "description" : "An HTTP-addressable URL pointing to the
    authentication scheme's specification.",
  "required" : false,
  "caseExact" : false,
  "mutability" : "readOnly",
  "returned" : "default",
  "uniqueness" : "none"
},
{
  "name" : "documentationUri",
  "type" : "reference",
  "referenceTypes" : ["external"],
  "multiValued" : false,
  "description" : "An HTTP-addressable URL pointing to the
    authentication scheme's usage documentation.",
  "required" : false,
  "caseExact" : false,
  "mutability" : "readOnly",
  "returned" : "default",
  "uniqueness" : "none"
}
]
}
],
},
```

```
{
  "id" : "urn:ietf:params:scim:schemas:core:2.0:ResourceType",
  "name" : "ResourceType",
  "description" : "Specifies the schema that describes a SCIM
    resource type",
  "attributes" : [
    {
      "name" : "id",
      "type" : "string",
      "multiValued" : false,
      "description" : "The resource type's server unique id.
        May be the same as the 'name' attribute.",
      "required" : false,
      "caseExact" : false,
      "mutability" : "readOnly",
      "returned" : "default",
      "uniqueness" : "none"
    },
    {
      "name" : "name",
      "type" : "string",
      "multiValued" : false,
      "description" : "The resource type name. When applicable,
        service providers MUST specify the name, e.g., 'User'.",
      "required" : true,
      "caseExact" : false,
      "mutability" : "readOnly",
      "returned" : "default",
      "uniqueness" : "none"
    },
    {
      "name" : "description",
      "type" : "string",
      "multiValued" : false,
      "description" : "The resource type's human-readable
        description. When applicable, service providers MUST
        specify the description.",
      "required" : false,
      "caseExact" : false,
      "mutability" : "readOnly",
      "returned" : "default",
      "uniqueness" : "none"
    }
  ],
}
```

```
{
  "name" : "endpoint",
  "type" : "reference",
  "referenceTypes" : ["uri"],
  "multiValued" : false,
  "description" : "The resource type's HTTP-addressable
    endpoint relative to the Base URL, e.g., '/Users'.",
  "required" : true,
  "caseExact" : false,
  "mutability" : "readOnly",
  "returned" : "default",
  "uniqueness" : "none"
},
{
  "name" : "schema",
  "type" : "reference",
  "referenceTypes" : ["uri"],
  "multiValued" : false,
  "description" : "The resource type's primary/base schema
    URI.",
  "required" : true,
  "caseExact" : true,
  "mutability" : "readOnly",
  "returned" : "default",
  "uniqueness" : "none"
},
{
  "name" : "schemaExtensions",
  "type" : "complex",
  "multiValued" : false,
  "description" : "A list of URIs of the resource type's schema
    extensions.",
  "required" : true,
  "mutability" : "readOnly",
  "returned" : "default",
  "subAttributes" : [
    {
      "name" : "schema",
      "type" : "reference",
      "referenceTypes" : ["uri"],
      "multiValued" : false,
      "description" : "The URI of a schema extension.",
      "required" : true,
      "caseExact" : true,
      "mutability" : "readOnly",
      "returned" : "default",
      "uniqueness" : "none"
    }
  ],
}
```

```
{
  "name" : "required",
  "type" : "boolean",
  "multiValued" : false,
  "description" : "A Boolean value that specifies whether
    or not the schema extension is required for the
    resource type. If true, a resource of this type MUST
    include this schema extension and also include any
    attributes declared as required in this schema extension.
    If false, a resource of this type MAY omit this schema
    extension.",
  "required" : true,
  "mutability" : "readOnly",
  "returned" : "default"
}
]
}
],
{
  "id" : "urn:ietf:params:scim:schemas:core:2.0:Schema",
  "name" : "Schema",
  "description" : "Specifies the schema that describes a
    SCIM schema",
  "attributes" : [
    {
      "name" : "id",
      "type" : "string",
      "multiValued" : false,
      "description" : "The unique URI of the schema.
        When applicable, service providers MUST specify the URI.",
      "required" : true,
      "caseExact" : false,
      "mutability" : "readOnly",
      "returned" : "default",
      "uniqueness" : "none"
    }
  ],
}
```

```
{
  "name" : "name",
  "type" : "string",
  "multiValued" : false,
  "description" : "The schema's human-readable name.  When
    applicable, service providers MUST specify the name,
    e.g., 'User'.",
  "required" : true,
  "caseExact" : false,
  "mutability" : "readOnly",
  "returned" : "default",
  "uniqueness" : "none"
},
{
  "name" : "description",
  "type" : "string",
  "multiValued" : false,
  "description" : "The schema's human-readable name.  When
    applicable, service providers MUST specify the name,
    e.g., 'User'.",
  "required" : false,
  "caseExact" : false,
  "mutability" : "readOnly",
  "returned" : "default",
  "uniqueness" : "none"
},
{
  "name" : "attributes",
  "type" : "complex",
  "multiValued" : true,
  "description" : "A complex attribute that includes the
    attributes of a schema.",
  "required" : true,
  "mutability" : "readOnly",
  "returned" : "default",
  "subAttributes" : [
    {
      "name" : "name",
      "type" : "string",
      "multiValued" : false,
      "description" : "The attribute's name.",
      "required" : true,
      "caseExact" : true,
      "mutability" : "readOnly",
      "returned" : "default",
      "uniqueness" : "none"
    }
  ],
}
```

```
{
  "name" : "type",
  "type" : "string",
  "multiValued" : false,
  "description" : "The attribute's data type.
    Valid values include 'string', 'complex', 'boolean',
    'decimal', 'integer', 'dateTime', 'reference'."
  "required" : true,
  "canonicalValues" : [
    "string",
    "complex",
    "boolean",
    "decimal",
    "integer",
    "dateTime",
    "reference"
  ],
  "caseExact" : false,
  "mutability" : "readOnly",
  "returned" : "default",
  "uniqueness" : "none"
},
{
  "name" : "multiValued",
  "type" : "boolean",
  "multiValued" : false,
  "description" : "A Boolean value indicating an
    attribute's plurality.",
  "required" : true,
  "mutability" : "readOnly",
  "returned" : "default"
},
{
  "name" : "description",
  "type" : "string",
  "multiValued" : false,
  "description" : "A human-readable description of the
    attribute.",
  "required" : false,
  "caseExact" : true,
  "mutability" : "readOnly",
  "returned" : "default",
  "uniqueness" : "none"
},
},
```

```
{
  "name" : "required",
  "type" : "boolean",
  "multiValued" : false,
  "description" : "A boolean value indicating whether or
    not the attribute is required.",
  "required" : false,
  "mutability" : "readOnly",
  "returned" : "default"
},
{
  "name" : "canonicalValues",
  "type" : "string",
  "multiValued" : true,
  "description" : "A collection of canonical values.  When
    applicable, service providers MUST specify the
    canonical types, e.g., 'work', 'home'.",
  "required" : false,
  "caseExact" : true,
  "mutability" : "readOnly",
  "returned" : "default",
  "uniqueness" : "none"
},
{
  "name" : "caseExact",
  "type" : "boolean",
  "multiValued" : false,
  "description" : "A Boolean value indicating whether or
    not a string attribute is case sensitive.",
  "required" : false,
  "mutability" : "readOnly",
  "returned" : "default"
},
}
```

```
{
  "name" : "mutability",
  "type" : "string",
  "multiValued" : false,
  "description" : "Indicates whether or not an attribute
    is modifiable.",
  "required" : false,
  "caseExact" : true,
  "mutability" : "readOnly",
  "returned" : "default",
  "uniqueness" : "none",
  "canonicalValues" : [
    "readOnly",
    "readWrite",
    "immutable",
    "writeOnly"
  ]
},
{
  "name" : "returned",
  "type" : "string",
  "multiValued" : false,
  "description" : "Indicates when an attribute is returned
    in a response (e.g., to a query).",
  "required" : false,
  "caseExact" : true,
  "mutability" : "readOnly",
  "returned" : "default",
  "uniqueness" : "none",
  "canonicalValues" : [
    "always",
    "never",
    "default",
    "request"
  ]
},
}
```

```
{
  "name" : "uniqueness",
  "type" : "string",
  "multiValued" : false,
  "description" : "Indicates how unique a value must be.",
  "required" : false,
  "caseExact" : true,
  "mutability" : "readOnly",
  "returned" : "default",
  "uniqueness" : "none",
  "canonicalValues" : [
    "none",
    "server",
    "global"
  ]
},
{
  "name" : "referenceTypes",
  "type" : "string",
  "multiValued" : true,
  "description" : "Used only with an attribute of type
    'reference'. Specifies a SCIM resourceType that a
    reference attribute MAY refer to, e.g., 'User'.",
  "required" : false,
  "caseExact" : true,
  "mutability" : "readOnly",
  "returned" : "default",
  "uniqueness" : "none"
},
```

```
{
  "name" : "subAttributes",
  "type" : "complex",
  "multiValued" : true,
  "description" : "Used to define the sub-attributes of a
    complex attribute.",
  "required" : false,
  "mutability" : "readOnly",
  "returned" : "default",
  "subAttributes" : [
    {
      "name" : "name",
      "type" : "string",
      "multiValued" : false,
      "description" : "The attribute's name.",
      "required" : true,
      "caseExact" : true,
      "mutability" : "readOnly",
      "returned" : "default",
      "uniqueness" : "none"
    },
    {
      "name" : "type",
      "type" : "string",
      "multiValued" : false,
      "description" : "The attribute's data type.
        Valid values include 'string', 'complex', 'boolean',
        'decimal', 'integer', 'dateTime', 'reference'."
      "required" : true,
      "caseExact" : false,
      "mutability" : "readOnly",
      "returned" : "default",
      "uniqueness" : "none",
      "canonicalValues" : [
        "string",
        "complex",
        "boolean",
        "decimal",
        "integer",
        "dateTime",
        "reference"
      ]
    }
  ],
}
```

```
{
  "name" : "multiValued",
  "type" : "boolean",
  "multiValued" : false,
  "description" : "A Boolean value indicating an
    attribute's plurality.",
  "required" : true,
  "mutability" : "readOnly",
  "returned" : "default"
},
{
  "name" : "description",
  "type" : "string",
  "multiValued" : false,
  "description" : "A human-readable description of the
    attribute.",
  "required" : false,
  "caseExact" : true,
  "mutability" : "readOnly",
  "returned" : "default",
  "uniqueness" : "none"
},
{
  "name" : "required",
  "type" : "boolean",
  "multiValued" : false,
  "description" : "A boolean value indicating whether or
    not the attribute is required.",
  "required" : false,
  "mutability" : "readOnly",
  "returned" : "default"
},
{
  "name" : "canonicalValues",
  "type" : "string",
  "multiValued" : true,
  "description" : "A collection of canonical values.  When
    applicable, service providers MUST specify the
    canonical types, e.g., 'work', 'home'.",
  "required" : false,
  "caseExact" : true,
  "mutability" : "readOnly",
  "returned" : "default",
  "uniqueness" : "none"
},
}
```

```
{
  "name" : "caseExact",
  "type" : "boolean",
  "multiValued" : false,
  "description" : "A Boolean value indicating whether or
    not a string attribute is case sensitive.",
  "required" : false,
  "mutability" : "readOnly",
  "returned" : "default"
},
{
  "name" : "mutability",
  "type" : "string",
  "multiValued" : false,
  "description" : "Indicates whether or not an
    attribute is modifiable.",
  "required" : false,
  "caseExact" : true,
  "mutability" : "readOnly",
  "returned" : "default",
  "uniqueness" : "none",
  "canonicalValues" : [
    "readOnly",
    "readWrite",
    "immutable",
    "writeOnly"
  ]
},
{
  "name" : "returned",
  "type" : "string",
  "multiValued" : false,
  "description" : "Indicates when an attribute is
    returned in a response (e.g., to a query).",
  "required" : false,
  "caseExact" : true,
  "mutability" : "readOnly",
  "returned" : "default",
  "uniqueness" : "none",
  "canonicalValues" : [
    "always",
    "never",
    "default",
    "request"
  ]
},
}
```


9. Security Considerations

9.1. Protocol

SCIM data is intended to be exchanged using the SCIM protocol. It is important when handling data to implement the security considerations outlined in Section 7 of [RFC7644].

9.2. Passwords and Other Sensitive Security Data

Passwords and other attributes related to security credentials are of an extremely sensitive nature and require special handling when transmitted or stored. While the SCIM protocol uses cleartext passwords for value assignment and equality-testing purposes, password values MUST NOT be stored in cleartext form.

Administrators should undertake industry best practices to protect the storage of credentials and in particular SHOULD follow recommendations outlined in Section 5.1.4.1 of [RFC6819]. These requirements include, but are not limited to, the following:

- o Provide injection attack countermeasures (e.g., by validating all inputs and parameters);
- o Credentials should not be stored in cleartext form;
- o Store credentials using an encrypted protection mechanism (e.g., hashing); and
- o Where possible, avoid passwords as the sole form of authentication, and consider using credentials that are based on asymmetric cryptography.

9.3. Privacy

The SCIM core schema defines attributes that are sensitive and may be considered personally identifying information (PII). These privacy considerations should be considered for extensions as well as the schema defined in this specification.

For the purposes of this specification, PII is defined as any attribute that may be used as a unique key to identify a person (e.g., "User"). Since other information may be used in combination to identify an individual, all attributes in SCIM are considered "sensitive" personal information. Consult regional jurisdictions to see if there are special considerations for the handling of personal information (e.g., PII).

Information should be shared on an as-needed basis. A SCIM client should limit information to what it believes a service provider requires, and a SCIM service provider should only accept information it needs. Clients and service providers should take into consideration that personal information is being conveyed across technical (e.g., protocol and applications), administrative (e.g., organizational, corporate), and jurisdictional boundaries. In particular, information security and privacy must be considered.

Security service level agreements for the handling of these attributes are beyond the scope of this document but are to be carefully considered by implementers and deploying organizations.

Please see the Privacy Considerations section of [RFC7644] for more protocol-specific considerations regarding the handling of SCIM information.

SCIM defines attributes such as "id", "externalId", and SCIM resource URIs, which cause new PII to be generated; this information is important to the way that the SCIM protocol identifies and locates resources. Where possible, it is suggested that service providers take the following remediations:

- o Where possible, assign and bind identifiers to specific tenants and/or clients. When multiple tenants are able to reference the same resource, they should do so via separate identifiers (id or externalId). This ensures that separate domains linked to the same information cannot perform identifier correlation.
- o In the case of "externalId", if multiple values are supported, use access control to restrict access to the client domain that assigned the "externalId" value.
- o Ensure that access to data is appropriately restricted to authorized parties with a "need to know".
- o When persisted, ensure that the appropriate protection mechanisms are in place to restrict access by unauthorized parties, including administrators or parties with access to backup data.

10. IANA Considerations

10.1. Registration of SCIM URN Sub-namespace and SCIM Registry

IANA has added an entry to the "IETF URN Sub-namespace for Registered Protocol Parameter Identifiers" registry and created a sub-namespace for the Registered Parameter Identifier as per [RFC3553]: "urn:ietf:params:scim".

To manage this sub-namespace, IANA has created the "System for Cross-domain Identity Management (SCIM) Schema URIs" registry, which is used to manage entries within the "urn:ietf:params:scim" namespace. The registry description is as follows:

- o Registry name: SCIM
- o Specification: this document (RFC 7643)
- o Repository: See Section 10.2
- o Index value: See Section 10.2

10.2. URN Sub-namespace for SCIM

SCIM schemas and SCIM messages utilize URIs to identify the schema in use or other relevant context. This section creates and registers an IETF URN Sub-namespace for use in the SCIM specifications and future extensions.

10.2.1. Specification Template

Namespace ID:

The Namespace ID "scim" has been assigned.

Registration Information:

Version: 1

Date: 2015-06-22

Declared registrant of the namespace:

Registering organization

The Internet Engineering Task Force

Designated contact

A designated expert will monitor the SCIM public mailing list, "scim@ietf.org".

Declaration of Syntactic Structure:

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

```
urn:ietf:params:scim:{type}:{name}{:other}
```

The keywords have the following meaning:

type

The entity type, which is either "schemas" or "api".

name

A required US-ASCII string that conforms to the URN syntax requirements (see [RFC2141]) and defines a major namespace of a schema used within SCIM (e.g., "core", which is reserved for SCIM specifications). The value MAY also be an industry name or organization name.

other

Any US-ASCII string that conforms to the URN syntax requirements (see [RFC2141]) and defines the sub-namespace (which MAY be further broken down in namespaces delimited by colons) as needed to uniquely identify a schema.

Relevant Ancillary Documentation:

None

Identifier Uniqueness Considerations:

The designated contact shall be responsible for reviewing and enforcing uniqueness.

Identifier Persistence Considerations:

Once a name has been allocated, it MUST NOT be reallocated for a different purpose. The rules provided for assignments of values within a sub-namespace MUST be constructed so that the meanings of values cannot change. This registration mechanism is not appropriate for naming values whose meanings may change over time.

As the SCIM specifications are updated and the SCIM protocol version is adjusted, a new registration will be made when significant changes are made -- for example, "urn:ietf:params:scim:schemas:core:1.0 (externally defined, not previously registered)" and "urn:ietf:params:scim:schemas:core:2.0".

Process of Identifier Assignment:

Identifiers with namespace type "schema" (e.g., "urn:ietf:params:scim:schemas") are assigned after the review of the assigned contact via the SCIM public mailing list, "scim@ietf.org", as documented in Section 10.3.

Namespaces with type "api" (e.g., "urn:ietf:params:scim:api") and "param" (e.g., "urn:ietf:params:scim:param") are reserved for IETF-approved SCIM specifications.

Process of Identifier Resolution:

The namespace is not currently listed with a Resolution Discovery System (RDS), but nothing about the namespace prohibits the future definition of appropriate resolution methods or listing with an RDS.

Rules for Lexical Equivalence:

No special considerations; the rules for lexical equivalence specified in [RFC2141] apply.

Conformance with URN Syntax:

No special considerations.

Validation Mechanism:

None specified.

Scope:

Global.

10.3. Registering SCIM Schemas

This section defines the process for registering new SCIM schemas with IANA in the "System for Cross-domain Identity Management (SCIM) Schema URIs" registry (see Section 10.1). A schema URI is used as a value in the "schemas" attribute (Section 3) for the purpose of distinguishing extensions used in a SCIM resource.

10.3.1. Registration Procedure

The IETF has created a mailing list, `scim@ietf.org`, which can be used for public discussion of SCIM schema proposals prior to registration. Use of the mailing list is strongly encouraged. The IESG has appointed a designated expert [RFC5226] who will monitor the `scim@ietf.org` mailing list and review registrations.

Registration of new "core" schemas (e.g., in the namespace "urn:ietf:params:scim:schemas:core") and "API" schemas (e.g., in the namespace "urn:ietf:params:scim:api") MUST be reviewed by the designated expert and published in an RFC. An RFC is REQUIRED for the registration of new value data types that modify existing properties. An RFC is also REQUIRED for registration of SCIM schema URIs that modify SCIM schema previously documented in an existing RFC. URNs within "urn:ietf:params:scim" but outside the above namespaces MAY be registered with a simple review (e.g., check for spam) by the designated expert on a first-come-first-served basis.

The registration procedure begins when a completed registration template, defined in the sections below, is sent to `scim@ietf.org` and `iana@iana.org`. Within two weeks, the designated expert is expected to tell IANA and the submitter of the registration whether the registration is approved, approved with minor changes, or rejected with cause. When a registration is rejected with cause, it can be resubmitted if the concerns listed in the cause are addressed.

Decisions made by the designated expert can be appealed to the IESG Applications Area Director, then to the IESG. They follow the normal appeals procedure for IESG decisions.

Once the registration procedure concludes successfully, IANA creates or modifies the corresponding record in the SCIM schema registry. The completed registration template is discarded.

An RFC specifying one or more new schema URIs MUST include the completed registration templates, which MAY be expanded with additional information. These completed templates are intended to go in the body of the document, not in the IANA Considerations section. The RFC SHOULD include any attributes defined.

10.3.2. Schema Registration Template

A SCIM schema URI is defined by completing the following template:

Schema URI: A unique URI for the SCIM schema extension.

Schema Name: A descriptive name of the schema extension (e.g., "Generic Device").

Intended or Associated Resource Type: A value defining the resource type (e.g., "Device").

Purpose: A description of the purpose of the extension and/or its intended use.

Single-value Attributes: A list and description of single-valued attributes defined, including complex attributes.

Multi-valued Attributes: A list and description of multi-valued attributes defined, including complex attributes.

10.4. Initial SCIM Schema Registry

The IANA has populated the "System for Cross-domain Identity Management (SCIM) Schema URIs" registry with the following registries for SCIM schema URIs, with pointers to appropriate reference documents. Note: The schema URIs listed below are broken into two lines for readability.

Schema URI	Name	Reference
urn:ietf:params:scim:schemas:core:2.0:User	User Resource	See Section 4.1
urn:ietf:params:scim:schemas:extension:enterprise:2.0:User	Enterprise User Extension	See Section 4.3
urn:ietf:params:scim:schemas:core:2.0:Group	Group Resource	See Section 4.2

SCIM Schema URIs for Data Resources

Schema URI	Name	Reference
urn:ietf:params:scim:schemas:core:2.0:ServiceProviderConfig	Service Provider Configuration Schema	See Section 5
urn:ietf:params:scim:schemas:core:2.0:ResourceType	Resource Type Configuration	See Section 6
urn:ietf:params:scim:schemas:core:2.0:Schema	Schema Definitions Schema	See Section 7

SCIM Server-Related Schema URIs

11. References

11.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC2141] Moats, R., "URN Syntax", RFC 2141, DOI 10.17487/RFC2141, May 1997, <<http://www.rfc-editor.org/info/rfc2141>>.
- [RFC3553] Mealling, M., Masinter, L., Hardie, T., and G. Klyne, "An IETF URN Sub-namespace for Registered Protocol Parameters", BCP 73, RFC 3553, DOI 10.17487/RFC3553, June 2003, <<http://www.rfc-editor.org/info/rfc3553>>.
- [RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, DOI 10.17487/RFC3629, November 2003, <<http://www.rfc-editor.org/info/rfc3629>>.
- [RFC3966] Schulzrinne, H., "The tel URI for Telephone Numbers", RFC 3966, DOI 10.17487/RFC3966, December 2004, <<http://www.rfc-editor.org/info/rfc3966>>.
- [RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, <<http://www.rfc-editor.org/info/rfc3986>>.
- [RFC4647] Phillips, A. and M. Davis, "Matching of Language Tags", BCP 47, RFC 4647, DOI 10.17487/RFC4647, September 2006, <<http://www.rfc-editor.org/info/rfc4647>>.
- [RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, DOI 10.17487/RFC4648, October 2006, <<http://www.rfc-editor.org/info/rfc4648>>.
- [RFC5234] Crocker, D., Ed., and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <<http://www.rfc-editor.org/info/rfc5234>>.
- [RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <<http://www.rfc-editor.org/info/rfc5280>>.

- [RFC5321] Klensin, J., "Simple Mail Transfer Protocol", RFC 5321, DOI 10.17487/RFC5321, October 2008, <<http://www.rfc-editor.org/info/rfc5321>>.
- [RFC5646] Phillips, A., Ed., and M. Davis, Ed., "Tags for Identifying Languages", BCP 47, RFC 5646, DOI 10.17487/RFC5646, September 2009, <<http://www.rfc-editor.org/info/rfc5646>>.
- [RFC6557] Lear, E. and P. Eggert, "Procedures for Maintaining the Time Zone Database", BCP 175, RFC 6557, DOI 10.17487/RFC6557, February 2012, <<http://www.rfc-editor.org/info/rfc6557>>.
- [RFC7159] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", RFC 7159, DOI 10.17487/RFC7159, March 2014, <<http://www.rfc-editor.org/info/rfc7159>>.
- [RFC7231] Fielding, R., Ed., and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content", RFC 7231, DOI 10.17487/RFC7231, June 2014, <<http://www.rfc-editor.org/info/rfc7231>>.
- [RFC7232] Fielding, R., Ed., and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Conditional Requests", RFC 7232, DOI 10.17487/RFC7232, June 2014, <<http://www.rfc-editor.org/info/rfc7232>>.
- [RFC7644] Hunt, P., Ed., Grizzle, K., Ansari, M., Wahlstroem, E., and C. Mortimore, "System for Cross-domain Identity Management: Protocol", RFC 7644, DOI 10.17487/RFC7644, September 2015, <<http://www.rfc-editor.org/info/rfc7644>>.

11.2. Informative References

- [ISO3166] International Organization for Standardization, "Codes for the representation of names of countries and their subdivisions - Part 1: Country codes", ISO 3166-1:2013, November 2013, <<http://www.iso.org>>.
- [Olson-TZ] Internet Assigned Numbers Authority, "IANA Time Zone Database", <<https://www.iana.org/time-zones>>.
- [PortableContacts] Smarr, J., "Portable Contacts 1.0 Draft C - Schema Only", August 2008, <<http://www.portablecontacts.net/draft-spec.html>>.

- [RFC2277] Alvestrand, H., "IETF Policy on Character Sets and Languages", BCP 18, RFC 2277, DOI 10.17487/RFC2277, January 1998, <<http://www.rfc-editor.org/info/rfc2277>>.
- [RFC4512] Zeilenga, K., Ed., "Lightweight Directory Access Protocol (LDAP): Directory Information Models", RFC 4512, DOI 10.17487/RFC4512, June 2006, <<http://www.rfc-editor.org/info/rfc4512>>.
- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, DOI 10.17487/RFC5226, May 2008, <<http://www.rfc-editor.org/info/rfc5226>>.
- [RFC6350] Perreault, S., "vCard Format Specification", RFC 6350, DOI 10.17487/RFC6350, August 2011, <<http://www.rfc-editor.org/info/rfc6350>>.
- [RFC6749] Hardt, D., Ed., "The OAuth 2.0 Authorization Framework", RFC 6749, DOI 10.17487/RFC6749, October 2012, <<http://www.rfc-editor.org/info/rfc6749>>.
- [RFC6819] Lodderstedt, T., Ed., McGloin, M., and P. Hunt, "OAuth 2.0 Threat Model and Security Considerations", RFC 6819, DOI 10.17487/RFC6819, January 2013, <<http://www.rfc-editor.org/info/rfc6819>>.
- [XML-Schema] Peterson, D., Gao, S., Malhotra, A., Sperberg-McQueen, C., and H. Thompson, "XML Schema Definition Language (XSD) 1.1 Part 2: Datatypes", April 2012, <<http://www.w3.org/TR/xmlschemall1-2/>>.

Acknowledgements

The editor would like to acknowledge the contribution and work of the editors of draft versions of this document:

Chuck Mortimore, Salesforce

Patrick Harding, Ping

Paul Madsen, Ping

Trey Drake, UnboundID

The SCIM Community would like to thank the following people for the work they've done in the research, formulation, drafting, editing, and support of this specification.

Morteza Ansari (morteza.ansari@cisco.com)

Sidharth Choudhury (schoudhury@salesforce.com)

Samuel Erdtman (samuel@erdtman.se)

Kelly Grizzle (kelly.grizzle@sailpoint.com)

Chris Phillips (cjphillips@gmail.com)

Erik Wahlstroem (erik.wahlstrom@nexusgroup.com)

Phil Hunt (phil.hunt@yahoo.com)

Special thanks to Joseph Smarr, whose excellent work on the Portable Contacts Specification [PortableContacts] provided a basis for the SCIM schema structure and text.

Authors' Addresses

Phil Hunt (editor)
Oracle Corporation

Email: phil.hunt@yahoo.com

Kelly Grizzle
SailPoint

Email: kelly.grizzle@sailpoint.com

Erik Wahlstroem
Nexus Technology

Email: erik.wahlstrom@nexusgroup.com

Chuck Mortimore
Salesforce.com

Email: cmortimore@salesforce.com