Development Guide

IM.PCF Interoperability

Author: Dieter Steding  
 Sophie Strecke

Creation Date: 27-11-2019

Last Updated: 23-09-2020

Document Ref: OC\_DE/300516442/DO.081/0005

Version: 1.0

**Approvals:**

|  |  |
| --- | --- |
| Bundeskriminalamt |  |
| Oracle Consulting |  |

logored Copy-No. \_\_\_\_\_

## Document Control

### Change Record

| Date | Author | Version | | Change Reference |
| --- | --- | --- | --- | --- |
|  |  |  |  | |
| 27-11-19 | Dieter Steding | 1.0 | No Previous Document | |
|  |  |  |  | |
|  |  |  |  | |
|  |  |  |  | |

### Reviewers

| Name | Position |
| --- | --- |
|  |  |
| René Donner | Project manager |
|  |  |
|  |  |
|  |  |

Contents

[Document Control ii](#_Toc57197819)

[Change Record ii](#_Toc57197820)

[Reviewers ii](#_Toc57197821)

[Overview 1](#_Toc57197822)

[Specification 1](#_Toc57197823)

[HTTP 1](#_Toc57197824)

[SCIM 2.0 1](#_Toc57197825)

[SCIM 1.1 1](#_Toc57197826)

[SCIM 1 (deprecated) 2](#_Toc57197827)

[Cloud Foundry API 2](#_Toc57197828)

[Compliance 3](#_Toc57197829)

[Interoperability 4](#_Toc57197830)

[User Account and Authentication Service (UAA) 4](#_Toc57197831)

[Endpoints 4](#_Toc57197832)

[Error Handling 4](#_Toc57197833)

[Data Types 5](#_Toc57197834)

[Attribute Filtering 6](#_Toc57197835)

[General 6](#_Toc57197836)

[Expected Result 6](#_Toc57197837)

[PCF Result 6](#_Toc57197838)

[Anomalies 7](#_Toc57197839)

[Expected Result 7](#_Toc57197840)

[PCF Result 7](#_Toc57197841)

[Retrieving Resources 8](#_Toc57197842)

[Expected Result 8](#_Toc57197843)

[PCF Result 8](#_Toc57197844)

[Cloud Controller API 9](#_Toc57197845)

[Resource Representation 9](#_Toc57197846)

[Compliance 9](#_Toc57197847)

[Inconvenience 9](#_Toc57197848)

[Retrieving Resources 10](#_Toc57197849)

[API Consistency 10](#_Toc57197850)

[Query Parameters 10](#_Toc57197851)

[Open Issues 11](#_Toc57197852)

[Identifier 11](#_Toc57197853)

[Consistency 11](#_Toc57197854)

## Overview

The System for Cross-domain Identity Management (SCIM) specification is designed to make managing user identities in cloud-based applications and services easier. The specification suite seeks to build 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 standard protocols. In essence: make it fast, cheap, and easy to move users in to, out of, and around the cloud.

### Specification

The following specifications are taken in account to depict the interoperability of the provided SCIM Service by Pivotal Cloud Foundry (PCF) User Account and Authorization (UAA).

1. HTTP
2. SCIM 2.0
3. SCIM 1.1
4. SCIM 1 (deprecated)
5. Cloud Foundry API

#### HTTP

Internet Engineering Task Force (IETF)

1. [Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content](https://tools.ietf.org/html/rfc7231)

#### SCIM 2.0

SCIM 2.0 is released as RFC7642, RFC7643 and RFC7644 under IETF in September 2015.

1. [RFC7643 - SCIM: Core Schema](https://tools.ietf.org/html/rfc7643)  
   The Core Schema provides a platform-neutral schema and extension model for representing users and groups.
2. [RFC7644 - SCIM: Protocol](https://tools.ietf.org/html/rfc7644)  
   The SCIM Protocol is an application-level, REST protocol for provisioning and managing identity data on the web.
3. [RFC7642 - SCIM: Definitions, Overview, Concepts, and Requirements](https://tools.ietf.org/html/rfc7642)  
   This document lists the user scenarios and use cases of System for Cross-domain Identity Management (SCIM).

#### SCIM 1.1

Second official release of the SCIM specification, released in July 2012. Compatible with 1.0 and contains cleanups and clarifications on issues found during interop testing.

1. [Core Schema](http://www.simplecloud.info/specs/draft-scim-core-schema-01.html)  
   The Core Schema provides a platform-neutral schema and extension model for representing users and groups in JSON and XML formats.
2. [REST API](http://www.simplecloud.info/specs/draft-scim-api-01.html)  
   The SCIM Protocol is an application-level, REST protocol for provisioning and managing identity data on the web.

#### SCIM 1 (deprecated)

First official release of the SCIM specification, released in December 2011.

1. [Core Schema](http://www.simplecloud.info/specs/draft-scim-core-schema-00.html)  
   The Core Schema provides a platform-neutral schema and extension model for representing users and groups in JSON and XML formats.
2. [REST API](http://www.simplecloud.info/specs/draft-scim-api-00.html)  
   The SCIM Protocol is an application-level, REST protocol for provisioning and managing identity data on the web.

#### Cloud Foundry API

1. [UAA API Version 74.24..0](https://docs.cloudfoundry.org/api/uaa/version/74.24.0/index.html#overview)
2. [REST API](https://apidocs.cloudfoundry.org/13.17.0/) Version 13.17.0 Cloud Controller API Version 2.153.0

## Compliance

|  |  |
| --- | --- |
|  | Important: |
| Any user can have multiple roles in multiple organizations and spaces |

| D:\tt\icon-restrict-16.png | Restrict: |
| --- | --- |
| Will it be a compliance issue in terms of SoD if a user is a manager of an organization and a developer in a space attached to the organization?  This constellation implies that the user have to be also the role user assigned on that organization |

## Interoperability

### User Account and Authentication Service (UAA)

#### Endpoints

|  |  |
| --- | --- |
|  | Important: |
| The SCIM protocol specifies well known endpoints and HTTP methods for managing Resources defined in the core schema; i.e., User and Group Resources correspond to /Users and /Groups respectively. Service Providers that support extended Resources SHOULD define Resource endpoints using the established convention; pluralize the Resource name defined in the extended schema by appending a 's'. Given there are cases where Resource pluralization is ambiguous; e.g., a Resource named 'person' is legitimately 'persons' and 'people' Consumers SHOULD discover Resource endpoints via the Schema Sub-Attribute 'endpoint'. |

| **Resource** | **SCIM 1/1.1** | **SCIM 2** | **Description** |
| --- | --- | --- | --- |
| **User** | /Users | /Users | Retrieve/Add/Modify Users |
| **Group** | /Groups | /Groups | Retrieve/Add/Modify Groups |
| **Service Provider Configuration** | /ServiceProviderConfigs | /ServiceProviderConfigs | Retrieve the Service Provider's Configuration |
| **Schema** | /Schemas | /Schemas | Retrieve a Resource's Schema |
| **Bulk** | /Bulk | /Bulk | Bulk modify Resources |

| D:\tt\icon-restrict-16.png | Restrict: |
| --- | --- |
| PCF response on Schema and Service Provider Configuration with empty collections |

| D:\tt\icon-warning-16.png | Workaround |
| --- | --- |
| Hardcoding of the of the schema request. |

| D:\tt\icon-note-16.png | Note: |
| --- | --- |
| Flexibility and maintainability is something else. |

#### Error Handling

|  |  |
| --- | --- |
|  | Important: |
| The SCIM Protocol uses the response status codes defined in HTTP to indicate operation success or failure. In addition to returning a HTTP response, code implementers MUST return the errors in the body of the response in the client requested format containing the error response and, per the HTTP specification, human-readable explanations. |

| D:\tt\icon-restrict-16.png | Restrict: |
| --- | --- |
| PCF provides in an error response the field as follows:   1. error\_description 2. error 3. message |

| D:\tt\icon-warning-16.png | Workaround |
| --- | --- |
| Extends the JSON parser to accept those fields that are outside of the standard. |

#### Data Types

|  |  |
| --- | --- |
|  | Important: |
| Attribute data types are derived from XML schema and unless otherwise specified are optional, modifiable by Consumers, and of type String. The JSON format defines a limited set of data types, hence, where appropriate, alternate JSON representations are defined below. SCIM extensions SHOULD not introduce new data types. |

| **Type** | **Description** |
| --- | --- |
| **String** | A sequence of characters as defined in [section 3.2.1](http://www.w3.org/TR/xmlschema-2/#string) of the XML Schema Datatypes Specification. A String attribute MAY specify a required data format. Additionally, when Canonical Values are specified Service Providers SHOULD conform to those values if appropriate, but MAY provide alternate String values to represent additional values. |
| **Boolean** | The literal "true" or "false" as specified in [section 3.2.2](http://www.w3.org/TR/xmlschema-2/#boolean) of the XML Schema Datatypes Specification. |
| **Decimal** | A real number with at least one digit to the left and right of the period as specified in [section 3.2.3](http://www.w3.org/TR/xmlschema-2/#boolean) of the XML Schema Datatypes Specification.  Values represented in JSON **must** conform to the XML constraints above and are represented as a [JSON Number](http://www.json.org/). |
| **Integer** | A Decimal number with no fractional digits as defined in [section 3.3.13](http://www.w3.org/TR/xmlschema-2/#integer) of the XML Schema Datatypes Specification.  Values represented in JSON **must** conform to the XML constraints above and are represented as a [JSON Number](http://www.json.org/). |
| **DateTime** | A dateTime (e.g. 2008-01-23T04:56:22Z) as specified in [section 3.2.7](http://www.w3.org/TR/xmlschema-2/#dateTime) of the XML Schema Datatypes Specification.  Values represented in JSON **must** conform to the XML constraints above and are represented as a [JSON String](http://www.json.org/). |
| **Binary** | The attribute value **must** be encoded as a valid xsd:base64Binary value as specified in [section 3.2.16](http://www.w3.org/TR/xmlschema-2/#base64Binary) of the XML Schema Datatypes Specification.  JSON are represented as a [JSON Object](http://www.json.org/)s. |

| D:\tt\icon-restrict-16.png | Restrict: |
| --- | --- |
| PCF use Long for types in the user resource. |

| D:\tt\icon-warning-16.png | Workaround |
| --- | --- |
| No idea at the time being |

#### Attribute Filtering

#### General

|  | Important: SCIM 1.1 – Core Schema |
| --- | --- |
| All operations share a common scheme for referencing simple and complex attributes. In general, attributes are uniquely identified by prefixing the attribute name with its schema URN separated by a colon (":") character; e.g., the core User resource attribute 'userName' is identified as "urn:ietf:params:scim:schemas:core:2.0:User:userName".  Complex attributes' sub-attributes are referenced via nested dot ('.') notation, i.e., {urn}:{Attribute name}.{Sub-Attribute name}. For example, the fully qualified path for a User's givenName is "urn:ietf:params:scim:schemas:core:2.0:User:name.givenName". All facets (URN, attribute, and sub-attribute name) of the fully encoded attribute name are case insensitive. |

Users and Group Resources are retrieved via opaque, unique URLs or via Query. Service Providers MAY choose to respond with a sub-set of Resource attributes, though MUST minimally return the Resource id and meta attributes.

| D:\tt\icon-restrict-16.png | Restrict: |
| --- | --- |
| PCF isn’t able to response properly on request  https://uaa.vm.pivotal.com/Users?attributes=meta.created |

##### Expected Result

{

"resources": [

{

"meta": {

"created":"2020-08-08T07:18:59.000Z"

}

, "id":"1599c2e5-8358-4e91-80c1-178819080f33"

}

]

}

##### PCF Result

{

"resources": [

{

"meta.created":"2020-08-08T07:18:59.000Z"

}

]

}

| D:\tt\icon-warning-16.png | Workaround |
| --- | --- |
| Flatten requested attributes to the parent so that any sub-attribute leads to the name of the complex attribute only. |

| D:\tt\icon-note-16.png | Note: |
| --- | --- |
| Performance will be less. |

#### Anomalies

|  | Important: UAA API Version 74.24..0 |
| --- | --- |
| Listing users supports SCIM filtering on the available attributes. When users are searched, we can return only selected amount of data using filtering. The attribute *groups* will cause the UAA to query the group memberships and include them in the result making the operation more expensive. The attribute *approvals* will cause the UAA to query the user approvals and include them in the result making the operation more expensive. |

| D:\tt\icon-restrict-16.png | Restrict: |
| --- | --- |
| That is a lie!  https://uaa.vm.pivotal.com/Users?attributes=id, groups |

##### Expected Result

{

"resources": [

{

"groups": [

{

"value": "ff75f8f0-0825-488c-b766-d8d2a8fab682"

, "display": "routing.router\_groups.read"

, "type": "DIRECT"

}

, {

"value": "d49cf130-1f29-4cd6-86b8-03537b9ac473",

"display": "cloud\_controller\_service\_permissions.read",

"type": "DIRECT"

}

, {

"value": "57f17bed-f1c4-422e-8f8b-4e3eb2735adc"

, "display": "approvals.me"

, "type": "DIRECT"

}

, ...

]

, "id":"1599c2e5-8358-4e91-80c1-178819080f33"

}

]

}

##### PCF Result

{

"resources": [

{

"groups": null

, "id":"1599c2e5-8358-4e91-80c1-178819080f33"

}

]

}

| D:\tt\icon-warning-16.png | Workaround |
| --- | --- |
| We have to query each embedded resource separately appropriately to the clients requested attributes what heavily decrease performance. |

| D:\tt\icon-note-16.png | Note: |
| --- | --- |
| Performance will be more than less. |

#### Retrieving Resources

|  | Important: |
| --- | --- |
| The following attributes control, which attributes SHALL be returned with a returned resource.   1. attributes 2. excludedAttributes   SCIM clients MAY use one of these two OPTIONAL parameters, which MUST be supported by SCIM service providers: |

Queries MAY be performed against a SCIM resource object, a resource type endpoint, or a SCIM server root. For example:

1. /Users/{id}
2. /Users
3. Groups
4. Groups/{id}

| D:\tt\icon-restrict-16.png | Restrict: |
| --- | --- |
| PCF ignores *attributes* and *excludedAttributes* query parameters if a resource is queried like /Users/{id}?attributes=id,meta |

##### Expected Result

{

"resources": [

{

"meta": {

"version":5

, "created":"2020-08-08T07:18:59.000Z"

, "lastModified":"2020-09-03T18:20:30.000Z"

}

, "id":"1599c2e5-8358-4e91-80c1-178819080f33"

}

]

}

##### PCF Result

Full-blown user resource.

| D:\tt\icon-warning-16.png | Workaround |
| --- | --- |
| Not applicable |

| D:\tt\icon-note-16.png | Note: |
| --- | --- |
| Hopefully we will not need such functionality. |

### Cloud Controller API

#### Resource Representation

##### Compliance

|  | Important: RFC7231 |
| --- | --- |
| HTTP does not define exactly how a PUT method affects the state of an origin server beyond what can be expressed by the intent of the user agent request and the semantics of the origin server response. It does not define what a resource might be, in any sense of that word, beyond the interface provided via HTTP. It does not define how resource state is "stored", nor how such storage might change as a result of a change in resource state, nor how the origin server translates resource state into representations. Generally speaking, all implementation details behind the resource interface are intentionally hidden by the server. |

|  | Important: RFC7231 |
| --- | --- |
| The fundamental difference between the POST and PUT methods is highlighted by the different intent for the enclosed representation. The target resource in a POST request is intended to handle the enclosed representation according to the resource's own semantics, whereas the enclosed representation in a PUT request is defined as replacing the state of the target resource. Hence, the intent of PUT is idempotent and visible to intermediaries, even though the exact effect is only known by the origin server. |

PCF use following representations in the Cloud Controller API:

PUT /v2/organizations/a0a4822a-3d97-47b3-adf6-337a8aa24722/managers/uaa-id-202

| D:\tt\icon-restrict-16.png | Restrict: |
| --- | --- |
| This semantics enforce a PUT method without a request body.  This behavior requires overriding the HTTP compliance checks in any client that use those API’s. |

| D:\tt\icon-warning-16.png | Workaround |
| --- | --- |
| Put something in the body |

##### Inconvenience

PCF use following representations in the Cloud Controller API:

PUT /v2/organizations/a0a4822a-3d97-47b3-adf6-337a8aa24722/managers/uaa-id-202

| D:\tt\icon-restrict-16.png | Restrict: |
| --- | --- |
| There is no chance to assign in bulk. |

| D:\tt\icon-warning-16.png | Workaround |
| --- | --- |
| Not applicable |

| D:\tt\icon-note-16.png | Note: |
| --- | --- |
| Hopefully we will not need such functionality. |

#### Retrieving Resources

##### API Consistency

The API of the Cloud Controller V2 is inconsistent in terms of applying query parameters.

The Space resource allows assignment and revocation of following roles

1. Manager
2. Auditor
3. Developer

To retrieve roles assigned to a user it can be assumed that applying a guid as a query criterion at this API will return all Space where permissions are granted.

| D:\tt\icon-restrict-16.png | Restrict: |
| --- | --- |
| The only accepted query parameter name at /v2/spaces is *developer\_guid*.  *manager\_guid* or *auditor\_guid* throwing exceptions about **The query parameter is invalid: manager\_guid** or **The query parameter is invalid: auditor\_guid** respectively. |

| D:\tt\icon-warning-16.png | Workaround |
| --- | --- |
| Not applicable. |

| D:\tt\icon-note-16.png | Note: |
| --- | --- |
| We will be able to provision but not able to reconcile. |

##### Query Parameters

| D:\tt\icon-restrict-16.png | Restrict: |
| --- | --- |
| A combination of query parameters is not applicable.  e.g.:  /v2/organizations?q=manager\_guid={id} or auditor\_guid={id}. |

| D:\tt\icon-warning-16.png | Workaround |
| --- | --- |
| We need to code the request of each possible role a user can have on a resource and query each of them with the allowed query parameters. |

| D:\tt\icon-note-16.png | Note: |
| --- | --- |
| Performance will be more than less. |

## Open Issues

### Identifier

|  | Important: |
| --- | --- |
| PCF generates for each resource an identifier (GUID) in the form **80da3c03-7cd1-4988-9015-05b6011ca8d9**. |

It has to be ensured by PCF that such identifiers are unique across multiple Foundations deployed on the same or different Iaas’s.

### Consistency

The API’s are working inconsistence in general

Rest API

| D:\tt\icon-restrict-16.png | Restrict: |
| --- | --- |
| In order for a space role to be set for a user, that user must first hold the role of "user" in the space's organization. The cf CLI hides this from you. But, if you turn on cf CLI trace prior to running cf set-space-role, you will see two PUTs are made, one to /v2/organizations/guid/users and one to /v2/spaces/guid/developers. |

But on the opposite

If you are creating a new user in UAA it gets provisioned some groups per default.

| D:\tt\icon-note-16.png | Note: |
| --- | --- |
| Funny stuff by a stoned developer team |