<html><head></head><body><pre style="word-wrap: break-word; white-space: pre-wrap;">

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 30, 2015

P. Hunt, Ed.
G. Wilson
Oracle
March 29, 2015

SCIM Password Management Extension
draft-hunt-scim-password-mgmt-00

Abstract

The System for Cross-Domain Identity Management (SCIM) specification is an HTTP based protocol that makes managing identities in multi-domain scenarios easier to support through a standardized services. SCIM provides extension points that enable new ResourceTypes and Schema Extensions to be defined. This specification defines a set of password and account status extensions for managing passwords and password usage (e.g. failures) and other related session data. The specification defines new ResourceTypes that enable management of passwords and account recovery functions.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 30, 2015.

Table of Contents

1. Introduction and Overview

The System for Cross-Domain Identity Management (SCIM) specification is an HTTP based protocol that makes managing identities in multi-domain scenarios easier to support through a standardized services. SCIM provides extension points that enable new ResourceTypes and Schema Extensions to be defined. This specification defines a set of password and account status extensions for managing passwords and tracking password usage (e.g. failures) and other related session data. The specification defines new resource types that enable management of passwords and account recovery functions.

A set of SCIM schema extensions that define:

- Password Schema Extension - Providing account password state (e.g. login attempts, successful login date, create date), policy, account locking, as well as challenge questions.

- Password Policy Schema - A new resource type that defines password policies that may be applied to resources that use passwords such as complexity requirements, expiry, lockout, and usage constraints.

A set of resource types are defined that enable password and password policy management:

- Password Policy
- Password Reset Request
- Password Validation Request
- Username Recovery Request

In the above list, the last 3 resource types are temporary resources that are used to convey requests that may update an identified target resource URI (e.g. a User). While these requests have a simple state transfer request/response relationship with a SCIM client, they may cause secondary effects by changing multiple attribute states in the target of the request. For example, setting a resource's password attribute involves validating password policy as well as checking and revising password history. There may be further service provider actions such as email confirmation that occur asynchronously from the SCIM client's perspective.

1.1. Notational 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]. These keywords are capitalized when used to unambiguously specify requirements of the protocol or application features and behavior that affect the interoperability and security of implementations. When these words are not capitalized, they are meant in their natural-language sense.

For purposes of readability examples are not URL encoded. Implementers MUST percent encode URLs as described in Section 2.1 of [RFC3986].

Throughout this documents all figures MAY contain spaces and extra line-wrapping for readability and space limitations. Similarly, some URI's contained within examples, have been shortened for space and readability reasons.

1.2. Definitions

[TBD]

2. Schema Extensions

2.1. Password Schema Extension

The following SCIM extension defines attributes used to manage account passwords within a service provider. The extension is applied to a "User" resource, but MAY be applied to other resources that use passwords. The password extension is identified using the following schema URI: "urn:ietf:params:scim:schemas:extension:account:2.0:Password".

The following Singular Attributes are defined:

passwordState
A Complex attribute that describes server provided attributes regarding the state of the resource's password.

createDate
A DateTime which specifies the date and time the current password was set.

cantChange
A Boolean indicating that the current password MAY NOT be changed and all other password expiry settings SHALL be ignored.

noExpiry
A Boolean indicating that password expiry policy will not be applied for the current resource.

lastSuccessfulLoginDate
A DateTime value indicating the last successful login date.

lastFailedLoginDate
A DateTime value indicating the last failed login date.

loginAttempts
An Integer value indicating the number of failed login attempts. The value is reset to 0 after a successful login.

resetAttempts
An Integer value indicating the number of password reset attempts.

passwordMustChange
A Boolean value that indicates that the subject password value MUST change at the next login. If not changed, typically the account is locked The value may be set indirectly when the subject's current password expires, or directly set by an administrator.

passwordPolicyUri
A URI reference value that indicates the address of a password policy that is used in relation to the current resource.

locked
A Complex attribute that indicates an account is locked (blocking new sessions). The following sub-attributes are defined:

reason
A number value indicating the reason for locking. Valid values are:

0 - locked due to failed login attempts.

1 - locked by an administrator.

2 - locked due to failed forgot password reset attempts

on
A Boolean value indicating the account is locked.

lockDate A DateTime indicating when the resource was locked.

duration An optional Integer indicating length of lockout in seconds.

The following Multi-valued Attributes are defined:

challenges
A Complex attribute describing challenge questions that may be used as a supplementary factor during login or during password management requests.

question
A String that represents a challenge question for which the corresponding response is defined.

response
A String that represents the subjects specified correct response to the corresponding challenge. The response MAY be compared case-sensitive or case-insensitive based on service provider policy.

passwordHistory
A writeOnly attribute that contains hashes of previous passwords associated with the SCIM resource. The number of passwords stored in this attribute is set by: "policy.passwordHistorySize". Persisted values MUST be securely hashed such that the clients may test if a clear-text value was previously used by looking for a matching hash within the array of values.

2.2. Password Policy

The following SCIM extension defines a new SCIM resource type known as "PasswordPolicy" and usually has an endpoint of "/PasswordPolicies". The password policy is identified using the following core schema URI: "urn:ietf:params:scim:schemas:core:2.0:policy:Password".

The following Single-value attributes are defined:

name
A String that is the name of the policy. Typically used for informational purposes (e.g. to display to the user).

description
A String that describes the current policy. Typically used for informational purposes (e.g. to display to a user).

maxLength
An Integer indicating the maximum password length (in characters). A value of 0 or no value SHALL indicate no maximum length restriction.

minLength
An Integer indicating the minimum password length (in characters). A value of 0 or no value SHALL indicate no minimum length

restriction.

minAlphas
An Integer indicating the minimum number of alphabetic characters in a password. A value of 0 or no value SHALL indicate no minimum length restriction.

minNumerals

An Integer indicating the minimum number of numeric characters in a password. A value of 0 or no value SHALL indicate no minimum length restriction.

minAlphaNumerals
An Integer indicating the minimum number of alphabetic or numeric characters in a password. A value of 0 or no value SHALL indicate no minimum length restriction.

minSpecialChars
An Integer indicating the minimum number of special characters in a password. A value of 0 or no value SHALL indicate no minimum length restriction.

maxSpecialChars
An Integer indicating the maximum number of special characters in a password. A value of 0 or no value SHALL indicate no maximum length restriction.

minUpperCase
An Integer indicating the minimum number of upper-case alphabetic characters in a password. A value of 0 or no value SHALL indicate no minimum length restriction.

minLowerCase
An Integer indicating the minimum number of lower-case alphabetic characters in a password. A value of 0 or no value SHALL indicate no minimum length restriction.

minUniqueChars
An Integer indicating the minimum number of unique characters in a password. A value of 0 or no value SHALL indicate no minimum restriction.

maxRepeatedChars
An Integer indicating the maximum number of repeated characters in a password. A value of 0 or no value SHALL indicate no restriction.

startsWithAlpha
A Boolean indicating that the password MUST being with an alphabetic character.

minUnicodeChars
[...not sure this makes sense. There are strict limitations on password values (must be Unicode UTF-8 processed by PRECIS)]

firstNameDisallowed

A Boolean indicating a sequence of characters matching the resource's "name.givenName" SHALL NOT be included in the password.

lastNameDisallowed
A Boolean indicating a sequence of characters matching the resource's "name.familyName" SHALL NOT be included in the password.

userNameDisallowed
A Boolean indicating a sequence of characters matching the resource's "userName" SHALL NOT be included in the password.

minPasswordAgeInDays
An Integer indicating the minimum age in days before the password MAY be changed.

warningAfterDays
An Integer indicating the number of days after which a password reset warning will be issued.

expiresAfterDays
An Integer indicating the numbers of days after which a password reset is required.

requiredChars
A String value whose contents indicates a set of characters that MUST appear, in any sequence, in a password value.

disallowedChars
A String value whose contents indicates a set of characters that SHALL NOT appear, in any sequence, in a password value.

disallowedSubStrings
A Multi-valued String indicating a set of Strings that SHALL NOT appear within a password value.

dictionaryLocation
A Reference value containing the URI of a dictionary of words not allowed to appear within a password value.

passwordHistorySize
An Integer indicating the number of passwords that will be kept in history that may not be used as a password.

maxIncorrectAttempts
An Integer representing the maximum number of failed logins before an account is locked.

lockOutDuration
An Integer indicating the number of minutes an account will be

locked after "maxIncorrectAttempts" exceeded.

challengesEnabled
A Boolean value indicating challenges MAY be used during authentication.

challengePolicy
A complex attribute that defines policy around challenges. It contains the following sub-attributes:

source An Integer indicating one of the following:

+ 0 - User Defined.

+ 1 - Admin Defined.

+ 2 - User and Admin Defined.

defaultQuestions A Multi-valued String attribute that contains one or more default question a subject may use when setting their challenge questions.

minQuestionCount An Integer indicating the minimum number of challenge questions a subject MUST answer when setting challenge question answers. A value of 0 or no value indicates no minimum.

minAnswerCount An Integer indicating the minimum number of challenge answers a subject MUST answer when attempting to reset their password via forgot password request.

allAtOnce A Boolean value. When true, the client UI will present all challengers in random order each time displayed. When false, the client UI will present one challenge question at a time where the subject MUST respond before the next is displayed.

minResponseLength An Integer indicating the minimum number of characters in a challenge response. No value or a value of 0 indicates no minimum length (effectively 1).

maxIncorrectAttempts An Integer indicates the maximum number of failed reset password attempts using challenges. If any challenges are wrong in a reset attempt, the user's "resetAttempts" counter will be incremented by 1. If "resetAttempts" is greater than "maxIncorrectAttempts", the

subject's account will be locked with a "locked.reason" value of 2 see Paragraph 3.

2.3. Management Requests

This extension defines a series of password and username management requests that are modeled as SCIM resource types. Each request acts as a "function" that MAY result in multiple changes to a designated resource (e.g. User). For example, setting a password involves the service provider validating the new password, updating the password, revising password history and resetting appropriate password state

values.

A management request is performed by doing a SCIM creation request for the associated management function resource type. Each request resource type has its own schema and resource type endpoint. The normal SCIM API rules apply to these requests. When a request is completed, a SCIM service provider MAY return the final state in the HTTP response, or it MAY return the location of the created request resource object that MAY be used for further processing. [TO BE CLARIFIED]

The following requests are supported and defined in the following sections:

- PasswordResetRequest
- PasswordValidateRequest
- UsernameValidateRequest
- GenerateUsernameRequest
- RecoverUsernameRequest

2.4. PasswordResetRequest

A password reset request is performed by performing a SCIM Create operation using HTTP POST to the endpoint for resource type "PasswordResetRequest" which is typically "/PasswordResetRequests". Upon receiving the request, the service provider, based on its own logic, validates the request, and based on its own internal logic subsequently resets the password of the resource identified by "userName". This request MAY be made anonymously (since the user is unable to authenticate) or through an authenticated web application component, who in turn may be unable to authenticate the user). [Add security considerations for this request]

Upon validating a request, the service provider may return either HTTP Status 200 (Ok), or it may return the request as a temporary resource that exists for a period of time (e.g. awaiting secondary approval or e-mail confirmation).

The core schema for a "PasswordResetRequest" is "urn:ietf:params:scim :schemas:core:2.0:password:PasswordResetRequest". The above schema can be used in several reset forms as described in the following two sections. This schema includes the following attributes:

userName
: A string value that matches the service provider unique identifier for the user.

challenges
: A Complex attribute describing challenge questions and responses that match the values found in the resource matched by the "userName" attribute.

  question

A String that represents a challenge question for which the corresponding response is defined.

response
A String that represents the subjects specified correct response to the corresponding challenge. The response MAY be compared case-sensitive or case-insensitive based on service provider policy.

2.4.1. Password Reset With Challenges

An anonymous (or authenticated web application) by providing a "userName" and the correct set of challenges and a new password value, MAY request that a service provider accept a requested "password" and set the "password" directly. The service provider might perform other secondary checks to confirm the requestors identity (email confirmation.


```
POST /PasswordResetRequests  HTTP/1.1
Host: example.com
Accept: application/json
Content-Type: application/json
Content-Length: ...
{
    "schemas":
        ["urn:ietf:params:scim:schemas:core:2.0:password:PasswordResetRequest"],
    "userName":
        "happyAlice",
    "challenges": [
    {
        "challenge":"what is your favorite color",
        "response":"red"
    },
    {
        "challenge":"what is name of your pet",
        "response":"pet"
    },
    {
        "challenge":"what is city of your birth",
        "response":"city"
    }],
    "password": "&lt;new password&gt;"
}
```

Upon processing a successful request, the SCIM service provider would respond with:

HTTP/1.1 200 OK

In the above example, the request is considered complete when response is returned. In this case, no permanent request object is created and so no HTTP Location value is returned. In some cases, the service provider MAY keep the request until workflow completes. If it wishes to allow clients to "poll" for status, it MAY create a resource and returns an HTTP Location in the response. [Is this needed?]

2.4.2. Reset With Email Confirmation

By providing only a "userName" value, an email conformation flow MAY be initiated that requires the subject to click on the link (to prove ownership of the known email) upon which the user is confirmed and the request is processed.


```
POST /PasswordResetRequests  HTTP/1.1
Host: example.com
Accept: application/json
Content-Type: application/json
Content-Length: ...

{
    "schemas":
        ["urn:ietf:params:scim:schemas:core:2.0:password:PasswordResetRequest"],
    "userName":
        "happyAlice"
}
```

Upon processing a successful request, the SCIM service provider SHALL respond with:

HTTP/1.1 200 OK

In the above example, it is expected that the User will be given a link to click on out-of-band. As such the current request completes with no further response. As with the Challenges variant, a service provider MAY provide an HTTP Location if the service provider intends to keep the request active until it is completed. [Is a persisted request needed?]

2.5. PasswordValidateRequest

A password validation request MAY be used to confirm that a proposed password value conforms to service provider policy and associated user policy and password state criteria (e.g. such as password history). A request is performed by performing a SCIM Create operation using HTTP POST to the endpoint for resource type "PasswordValidateRequest" which is typically "/PasswordValidateRequests". Upon receiving the request, the service provider, based on its own logic and any associated password policy for the resource, validates the provided password. [can this be made anonymously?]

Upon validating a request, the service provider may returns either

HTTP Status 200 (Ok), or it may return HTTP Status 400 indicating the password is unacceptable. [NOTE: should there be a scimType and/or description describing a standardized reason for failure such as: history, tooShort, tooLong, missingSpecialChar, etc etc.

The core schema for a "PasswordValidateRequest" is "urn:ietf:params:scim:schemas:core:2.0:password:PasswordValidateRequest". This schema includes the following attributes:

$ref

A reference value that contains a URI that points to the resource (e.g. User) against which the proposed password is to be validated as an acceptable password.

password
A string value containing the requested password value for which validation is requested.

The following is a non-normative example validation request. The example has been altered for clarity:

```
POST /PasswordValidateRequests  HTTP/1.1
Host: example.com
Accept: application/json
Content-Type: application/json
Content-Length: ...
{
    "schemas":
        ["urn:ietf:params:scim:schemas:core:2.0:password:PasswordValidateRequest"],
    "$ref": "/Users/2819c223-7f76-453a-919d-413861904646",
    "password":"someG00Idea!"
}
```

A successful response looks similar to the following non-normative example:

```
HTTP/1.1 200 OK
```

2.6. UsernameValidateRequest

A username validation request MAY be used to confirm that a proposed username value conforms to service provider policy and associated user policy as well as uniqueness. A request is performed by performing a SCIM Create operation using HTTP POST to the endpoint for resource type "UsernameValidateRequest" which is typically "/UsernameValidateRequests". Upon receiving the request, the service provider, tests for uniqueness and any associated formatting policy and validates the provided username.

Upon validating a request, the service provider may returns either HTTP Status 200 (Ok), or it may return HTTP Status 400 indicating the password is unacceptable. [NOTE: should there be a scimType and/or description describing a standardized reason for failure such as: history, tooShort, tooLong, missingSpecialChar, etc etc.

The core schema for a "UsernameValidateRequest" is "urn:ietf:params:scim:schemas:core:2.0:password:UsernameValidateRequest". This schema

includes the following attributes:

$ref
A reference value that contains a URI that points to the resource (e.g. User) against which the proposed userName value is to be validated as an acceptable.

userName
A string value containing the requested userName value for which validation is requested.

The following is a non-normative example validation request. The example has been altered for clarity:

```
POST /UsernameValidaeRequests  HTTP/1.1
Host: example.com
Accept: application/json
Content-Type: application/json
Content-Length: ...
{
    "schemas":
        ["urn:ietf:params:scim:schemas:core:2.0:password:UsernameValidateRequest"],
    "$ref": "/Users/2819c223-7f76-453a-919d-413861904646",
    "userName":"susieQ"
}
```

A successful response looks similar to the following non-normative example:

HTTP/1.1 200 OK

2.7. UsernameGenerateRequest

A username generation request MAY be used to request an automatically generated userName that conforms to service provider policy and uniqueness requirements. A request is performed by performing a SCIM Create operation using HTTP POST to the endpoint for resource type "UsernameGenerateRequest" which is typically "/UsernameGenerateRequests". Upon receiving the request, the service provider, generates a unique userName and returns it in a response.

The core schema for a "UsernameGenerateRequest" is "urn:ietf:params:scim:schemas:core:2.0:password:UsernameGenerateRequest". This schema includes the following attributes:

$ref
An operational reference value that contains a URI that points to the resource (e.g. User) against which existing resource's "name" attribute MAY be used to generate a userName value. When the $ref attribute is used, the generate request MUST be authenticated.

userName
   A string value that is returned in the server's response that
   contains a generated userName value. The generated userName is
   not reserved and is guaranteed on first-come-first-served basis by
   a subsequent SCIM creation or modify request.

name
   An optional complex attribute containing the components of the
   user's name against which a userName value is to be generated.
   This attribute MAY be typically used as part of an anonymous
   userName generation request during a user registration dialog.

   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.").

   The following is a non-normative example userName generation request.
   The example has been altered for clarity:

```
POST /UsernameGenerateRequests  HTTP/1.1
Host: example.com
Accept: application/json
Content-Type: application/json
Content-Length: ...
{
```

```
    "schemas":
        ["urn:ietf:params:scim:schemas:core:2.0:password:UsernameGenerateRequest"],
    "name": {
        "formatted": "Ms. Barbara J Doe III",
        "familyName": "Doe",
        "givenName": "Barbara",
        "middleName": "Jane",
        "honorificSuffix": "III"
    }
}
```

A successful response looks similar to the following non-normative example:

```
HTTP/1.1 200 OK
{
    "userName": "barbara.doe",
}
```

2.8. UsernameRecoverRequest

A userName recovery request MAY be used to look up a userName based on a provided email address. The provided email address may be matched against any value of an existing resource's "emails" attribute. A request is performed by performing a SCIM Create operation using HTTP POST to the endpoint for resource type "UsernameRecoverRequest" which is typically "/UsernameRecoverRequests". Upon receiving the request, the service provider, generates a unique userName and returns it in a response.

The core schema for a "UsernameRecoverRequest" is "urn:ietf:params:scim:schemas:core:2.0:password:UsernameRecoverRequest". This schema includes the following attributes:

email
: A string value containing an email address that is to be matched against an existing resource's "emails" attribute.

userName
: A string value provided in response to a request which is the unique userName that corresponds to the recovery request.

The following is a non-normative example userName recovery request. The example has been altered for clarity:

```
POST /UsernameRecoverRequests  HTTP/1.1
Host: example.com
Accept: application/json
Content-Type: application/json
Content-Length: ...
{
  "schemas":
    ["urn:ietf:params:scim:schemas:core:2.0:password:UsernameRecoverRequest"],
   "email": "bdoe@example.com"
}
```

A successful response looks similar to the following non-normative

example:

```
HTTP/1.1 200 OK
{
    "userName": "barbara.doe",
}
```

[Note: it would be more secure not to return the userName in the response and instead the service provider should send an email confirmation]

3. Schemas Representation

This section provides a JSON representation of the schema extensions in this draft. [TODO follow format of Sec 8.7 of core schema draft]

3.1. Password Extension

The following is a representation of the password state extension "urn:ietf:params:scim:schemas:extension:account:2.0:Password" that is used to extend a User resource.

```
 {
  "id" :
    "urn:ietf:params:scim:schemas:extension:account:2.0:Password",
  "name" : "Password Management Schema Extension",
  "description" : "This extension defines attributes used to manage
    account passwords within a service provider.  The extension is
    typically applied to a User resource, but MAY be applied to
    other resources that use passwords.",
```



```
  "attributes" : [
    {
      "name" : "passwordState",
      "type" : "complex",
      "multiValued" : false,
      "description" : "A Complex attribute that describes server
        provided attributes regarding the state of the resource's
        password.",
      "required" : true,
      "returned" : "default",
      "mutability" : "readWrite",
      "subAttributes" : [
        {
          "name" : "createDate",
          "type" : "dateTime",
          "multiValued" : false,
          "description" : "A DateTime which specifies the date and
            time the current password was set.",
          "required" : false,
          "mutability" : "readWrite",
          "returned" : "default"
        },
        {
          "name" : "cantChange",
          "type" : "boolean",
          "multiValued" : false,
          "description" : "A Boolean indicating that the current
```

```
          password MAY NOT be changed and all other password expiry
          settings SHALL be ignored",
        "required" : false,
        "mutability" : "readWrite",
        "returned" : "default"
      },
      {
        "name" : "noExpiry",
        "type" : "boolean",
        "multiValued" : false,
        "description" : "A Boolean indicating that password expiry
          policy will not be applied for the current resource.",
        "required" : false,
        "mutability" : "readWrite",
        "returned" : "default"
      },
      {
        "name" : "lastSuccessfulLoginDate",
        "type" : "dateTime",
        "multiValued" : false,
        "description" : "A DateTime value indicating the last
```



```
          successful login date.",
        "required" : false,
        "mutability" : "readWrite",
        "returned" : "default"
      },
      {
        "name" : "lastFailedLoginDate",
        "type" : "dateTime",
        "multiValued" : false,
        "description" : "A DateTime value indicating the last
          failed login date.",
        "required" : false,
        "mutability" : "readWrite",
        "returned" : "default"
      },
      {
        "name" : "loginAttempts",
        "type" : "integer",
        "multiValued" : false,
        "description" : "An Integer value indicating the number of
          failed login attempts. The value is reset to 0 after a
          successfull login.",
        "required" : false,
        "mutability" : "readOnly",
        "returned" : "default"
      },
      {
        "name" : "resetAttempts",
        "type" : "integer",
        "multiValued" : false,
        "description" : "An Integer value indicating the number of
          password reset attempts. The value is reset to 0 after
          successful reset.",
        "required" : false,
        "mutability" : "readOnly",
        "returned" : "default"
```

```
        },
        {
          "name" : "passwordMustChange",
          "type" : "boolean",
          "multiValued" : false,
          "description" : "A Boolean value that indicates that the
            subject password value MUST change at the next login. If
            not changed, typically the account is locked The value
            may be set indirectly when the subject's current password
            expires, or directly set by an administrator.",
          "required" : false,
          "mutability" : "readWrite",
```



```
          "returned" : "default"
        }
      ]
    },
    {
      "name" : "passwordPolicyUrl",
      "type" : "reference",
      "referenceTypes" : ["PasswordPolicy"],
      "multiValued" : false,
      "description" : "A URI reference value that indicates the
        address of a password policy that is used in relation to the
        current resource.",
      "required" : false,
      "caseExact" : false,
      "mutability" : "readWrite",
      "returned" : "default",
      "uniqueness" : "none"
    },
    {
      "name" : "locked",
      "type" : "complex",
      "multiValued" : false,
      "description" : "A Complex attribute that indicates an account
        is locked (blocking new sessions).",
      "required" : false,
      "returned" : "default",
      "mutability" : "readWrite",
      "subAttributes" : [
        {
          "name" : "reason",
          "type" : "integer",
          "multiValued" : false,
          "description" : "A number value indicating the reason for
            locking. Valid values are: 0 - failed attempts. 1 - admin
            lock. 2 - reset attempts",
          "required" : true,
          "mutability" : "readWrite",
          "returned" : "default"
        },
        {
          "name" : "on",
          "type" : "boolean",
          "multiValued" : false,
          "description" :
            "A Boolean value indicating the account is locked.",
```

```
            "required" : true,
            "mutability" : "readWrite",
            "returned" : "default"
```



```
          },
          {
            "name" : "lockDate",
            "type" : "dateTime",
            "multiValued" : false,
            "description" : "A DateTime which specifies the date and
              time the current resource was locked.",
            "required" : false,
            "mutability" : "readWrite",
            "returned" : "default"
          }
        ]
      },
      {
        "name" : "challenges",
        "type" : "complex",
        "multiValued" : true,
        "description" : "A Complex attribute describing challenge
          questions that may be used as a supplementary factor during
          login or during password management requests.",
        "required" : false,
        "returned" : "default",
        "mutability" : "readWrite",
        "subAttributes" : [
          {
            "name" : "question",
            "type" : "string",
            "multiValued" : false,
            "description" : "A String that represents a challenge
              question for which the corresponding response is
              defined.",
            "required" : true,
            "caseExact" : true,
            "mutability" : "readWrite",
            "returned" : "default",
            "uniqueness" : "none"
          },
          {
            "name" : "response",
            "type" : "string",
            "multiValued" : false,
            "description" : "A String that represents the subjects
              specified correct response to the corresponding
              challenge.",
            "required" : true,
            "caseExact" : false,
            "mutability" : "readWrite",
            "returned" : "default",
```

```
            "uniqueness" : "none"
          }
        ]
      },
      {
        "name" : "passwordHistory",
        "type" : "string",
        "multiValued" : true,
        "description" : "A writeOnly attribute that contains hashes of
          previous passwords associated with the SCIM resource.",
        "required" : false,
        "caseExact" : true,
        "mutability" : "writeOnly",
        "returned" : "never",
        "uniqueness" : "none"
      }
    ]
  }
```


3.2. Password Policy Schema

The following is a representation of the password policy resource type extension "urn:ietf:params:scim:schemas:core:2.0:policy:Password" that is used to define a PasswordPolicy resource.

```
  {
    "id" :
      "urn:ietf:params:scim:schemas:core:2.0:policy:Password",
    "name" : "Password Policy Schema",
    "description" : "This extension defines attributes for a password
      policy.",
    "attributes" : [
      {
        "name" : "name",
        "type" : "string",
        "multiValued" : false,
        "description" : "A String that is the name of the policy.
          Typically used for informational purposes (e.g. to display
          to the user)",
        "required" : true,
        "caseExact" : false,
        "mutability" : "readWrite",
        "returned" : "default",
        "uniqueness" : "none"
      },
```



```
      {
        "name" : "description",
        "type" : "string",
        "multiValued" : false,
        "description" : "A String that describes the current policy.
          Typically used for informational purposes (e.g. to display
          to a user).",
```

```
      "required" : false,
      "caseExact" : false,
      "mutability" : "readWrite",
      "returned" : "default",
      "uniqueness" : "none"
    },
    {
      "name" : "maxLength",
      "type" : "integer",
      "multiValued" : false,
      "description" : "Maximum password length in characters.",
      "required" : false,
      "mutability" : "readWrite",
      "returned" : "default"
    },
    {
      "name" : "minLength",
      "type" : "integer",
      "multiValued" : false,
      "description" : "Minimum password length in characters.",
      "required" : false,
      "mutability" : "readWrite",
      "returned" : "default"
    },
    {
      "name" : "minAlphas",
      "type" : "integer",
      "multiValued" : false,
      "description" : "Minimum number of alpha chcars.",
      "required" : false,
      "mutability" : "readWrite",
      "returned" : "default"
    },
     {
      "name" : "minNumerals",
      "type" : "integer",
      "multiValued" : false,
      "description" : "Minimum number of numeric characters.",
      "required" : false,
      "mutability" : "readWrite",
      "returned" : "default"
```



```
    },
     {
      "name" : "maxLength",
      "type" : "integer",
      "multiValued" : false,
      "description" : "Maximum password length in characters.",
      "required" : false,
      "mutability" : "readWrite",
      "returned" : "default"
    },
    {
      "name" : "minAlphaNumerals",
      "type" : "integer",
      "multiValued" : false,
      "description" : "Minimum num of alphas and numeric chars.",
      "required" : false,
```

```
      "mutability" : "readWrite",
      "returned" : "default"
    },
    {
      "name" : "minSpecialChars",
      "type" : "integer",
      "multiValued" : false,
      "description" : "Minimum num of special chars.",
      "required" : false,
      "mutability" : "readWrite",
      "returned" : "default"
    },
    {
      "name" : "maxSpecialChars",
      "type" : "integer",
      "multiValued" : false,
      "description" : "Maximum number of special chars.",
      "required" : false,
      "mutability" : "readWrite",
      "returned" : "default"
    },
    {
      "name" : "minUpperCase",
      "type" : "integer",
      "multiValued" : false,
      "description" : "Minimum num of upper case chars.",
      "required" : false,
      "mutability" : "readWrite",
      "returned" : "default"
    },
    {
      "name" : "minLowerCase",
```



```
      "type" : "integer",
      "multiValued" : false,
      "description" : "Minimum num of lower case chars.",
      "required" : false,
      "mutability" : "readWrite",
      "returned" : "default"
    },
    {
      "name" : "minUnique",
      "type" : "integer",
      "multiValued" : false,
      "description" : "Minimum num of unique chars.",
      "required" : false,
      "mutability" : "readWrite",
      "returned" : "default"
    },
    {
      "name" : "maxRepeatChars",
      "type" : "integer",
      "multiValued" : false,
      "description" : "Max num of repeated chars.",
      "required" : false,
      "mutability" : "readWrite",
      "returned" : "default"
    },
```

```
    {
      "name" : "startsWithAlphas",
      "type" : "boolean",
      "multiValued" : false,
      "description" : "Indicates password must begin with alpha char",
      "required" : false,
      "mutability" : "readWrite",
      "returned" : "default"
    },
    {
      "name" : "minUnicodeChars",
      "type" : "integer",
      "multiValued" : false,
      "description" : "[TO BE DISCUSSED]",
      "required" : false,
      "mutability" : "readWrite",
      "returned" : "default"
    },
    {
      "name" : "firstNameDisallowed",
      "type" : "boolean",
      "multiValued" : false,
      "description" : "Indicates a sequence of characters matching
```



```
        the resource's name.givenName SHALL NOT be included in the
        password",
      "required" : false,
      "mutability" : "readWrite",
      "returned" : "default"
    },
    {
      "name" : "lastNameDisallowed",
      "type" : "boolean",
      "multiValued" : false,
      "description" : "Indicates a sequence of characters matching
        the resource's name.familyName SHALL NOT be included in the
        password",
      "required" : false,
      "mutability" : "readWrite",
      "returned" : "default"
    },
    {
      "name" : "userNameDisallowed",
      "type" : "boolean",
      "multiValued" : false,
      "description" : "Indicates a sequence of characters matching
        the resource's userName SHALL NOT be included in the
        password",
      "required" : false,
      "mutability" : "readWrite",
      "returned" : "default"
    },
    {
      "name" : "minPasswordAgeInDays",
      "type" : "integer",
      "multiValued" : false,
      "description" : "An Integer indicating the minimum age in days
        before the password MAY be changed.",
```

```
      "required" : false,
      "mutability" : "readWrite",
      "returned" : "default"
    },
    {
      "name" : "warningAfterDays",
      "type" : "integer",
      "multiValued" : false,
      "description" : "An Integer indicating the number of days after
        which a password reset warning will be issued.",
      "required" : false,
      "mutability" : "readWrite",
      "returned" : "default"
    },
```



```
    {
      "name" : "expiresAfterDays",
      "type" : "integer",
      "multiValued" : false,
      "description" : "An Integer indicating the numbers of days
        after which a password reset is required.",
      "required" : false,
      "mutability" : "readWrite",
      "returned" : "default"
    },
    {
      "name" : "requiredChars",
      "type" : "string",
      "multiValued" : false,
      "description" : "A String value whose contents indicates a set
        of characters that MUST appear, in any sequence, in a
        password value.",
      "required" : false,
      "caseExact" : true,
      "mutability" : "readWrite",
      "returned" : "never",
      "uniqueness" : "none"
    },
    {
      "name" : "disallowedChars",
      "type" : "string",
      "multiValued" : false,
      "description" : "A String value whose contents indicates a set
        of characters that SHALL NOT appear, in a password value.",
      "required" : false,
      "caseExact" : true,
      "mutability" : "readWrite",
      "returned" : "never",
      "uniqueness" : "none"
    },
    {
      "name" : "disallowedSubstrings",
      "type" : "string",
      "multiValued" : true,
      "description" : "A set of strings that SHALL not appear in a
        password value.",
      "required" : false,
      "caseExact" : true,
```

```
      "mutability" : "readWrite",
      "returned" : "never",
      "uniqueness" : "none"
    },
    {
```



```
      "name" : "disctionaryLocation",
      "type" : "reference",
      "referenceTypes" : ["reference"],
      "multiValued" : false,
      "description" : "A Reference value containing the URI of a
        dictionary of words not allowed to appear within a password
        value.",
      "required" : false,
      "caseExact" : false,
      "mutability" : "readWrite",
      "returned" : "default",
      "uniqueness" : "none"
    },
    {
      "name" : "passwordHistorySize",
      "type" : "integer",
      "multiValued" : false,
      "description" : "An Integer indicating the number of passwords
        that will be kept in history that may not be used as a
        password.",
      "required" : false,
      "mutability" : "readWrite",
      "returned" : "default"
    },
    {
      "name" : "maxIncorrectAttempts",
      "type" : "integer",
      "multiValued" : false,
      "description" : "An Integer representing the maximum number of
        failed logins before an account is locked.",
      "required" : false,
      "mutability" : "readWrite",
      "returned" : "default"
    },
    {
      "name" : "lockOutDuration",
      "type" : "integer",
      "multiValued" : false,
      "description" : "An integer indicating the number of minutes
        an account will be locked after maxIncorrectAttempts
        exceeded.",
      "required" : false,
      "mutability" : "readWrite",
      "returned" : "default"
    },
    {
      "name" : "challengesEnabled",
      "type" : "boolean",
```

```
      "multiValued" : false,
      "description" : "Indicates whether challenges may be used
        during authentication.",
      "required" : false,
      "mutability" : "readWrite",
      "returned" : "default"
    },
    {
      "name" : "challengePolicy",
      "type" : "complex",
      "multiValued" : false,
      "description" : "A complex attribute that defines policy around
        challenges.",
      "required" : true,
      "returned" : "default",
      "mutability" : "readWrite",
      "subAttributes" : [
        {
          "name" : "source",
          "type" : "integer",
          "multiValued" : false,
          "description" : "A number value indicating the source for
            challenges. Valid values are: 0 - user. 1 - admin
            defined. 2 - both",
          "required" : true,
          "mutability" : "readWrite",
          "returned" : "default"
        },
        {
          "name" : "defaultQuestions",
          "type" : "string",
          "multiValued" : true,
          "description" : "A Multi-valued String attribute that
            contains one or more default question a subject may use
            when setting their challenge questions",
          "required" : false,
          "caseExact" : false,
          "mutability" : "readWrite",
          "returned" : "default",
          "uniqueness" : "none"
        },
        {
          "name" : "minQuestionCount",
          "type" : "integer",
          "multiValued" : false,
          "description" : "An Integer indicating the minimum number
            of challenge questions a subject MUST answer when setting
            challenge question answers.  A value of 0 or no value
```



```
            indicates no minimum.",
          "required" : true,
          "mutability" : "readWrite",
          "returned" : "default"
        },
```

```
{
  "name" : "minAnswerCount",
  "type" : "integer",
  "multiValued" : false,
  "description" : "An Integer indicating the mimimum number
    of challenge answers a subject MUST answer when
    attempting to reset their password via forgot password
    request.",
  "required" : true,
  "mutability" : "readWrite",
  "returned" : "default"
},
{
  "name" : "allAtOnce",
  "type" : "boolean",
  "multiValued" : false,
  "description" : "When true, the client UI will present
    all challengers in random order each time displayed.
    When false, the client UI will present one challenge
    question at a time where the subject MUST respond before
    the next is displayed.",
  "required" : true,
  "mutability" : "readWrite",
  "returned" : "default"
},
{
  "name" : "minResponseLength",
  "type" : "integer",
  "multiValued" : false,
  "description" : "An Integer indicating the minimum number
    of chars in a challenge response.  No value or a value
    of 0 indicates no minimum length (effectively 1)",
    "required" : true,
    "mutability" : "readWrite",
    "returned" : "default"
  },
  {
    "name" : "maxIncorrectAttempts",
    "type" : "integer",
    "multiValued" : false,
    "description" : "An Integer indicates the maximum number of
      failed reset password attempts using challenges. If any
      challenges are wrong in a reset attempt, the user's
```



```
      resetAttempts counter will be incremented by 1.  If
      resetAttempts is greater than maxIncorrectAttempts, the
      subject's account will be locked with a locked.reason
      value.",
    "required" : true,
    "mutability" : "readWrite",
    "returned" : "default"
  }
 ]
 }
 ]
}
```

Password Policy Schema

3.3. Request Schemas

The following are the schemas for all password request resource types returned by the "/Schemas" endpoint:

```
[
  {
    "id" :
"urn:ietf:params:scim:schemas:core:2.0:password:PasswordResetRequest",
    "name" : "Password Reset Request",
    "description" : "Used to submit a password reset request for a
    specific userName. Before resetting a secondary confirmation is
    completed.",
    "attributes" : [
      {
        "name" : "userName",
        "type" : "string",
        "multiValued" : false,
        "description" : "A string value that matches the service provider
        unique identifier for the user.",
        "required" : true,
        "caseExact" : false,
        "mutability" : "readWrite",
        "returned" : "default",
        "uniqueness" : "none"
      },
      {
        "name" : "challenges",
        "type" : "complex",
        "multiValued" : true,
        "description" : "A Complex attribute describing challenge
        questions and responses that match the values found in the
        resource matched by the userName attribute.",
```



```
        "required" : false,
        "returned" : "default",
        "mutability" : "readWrite",
        "subAttributes" : [
          {
            "name" : "question",
            "type" : "string",
            "multiValued" : false,
            "description" : "A String that represents a challenge
               question for which the corresponding response is
               defined.",
            "required" : true,
            "caseExact" : true,
            "mutability" : "readWrite",
            "returned" : "default",
            "uniqueness" : "none"
          },
          {
            "name" : "response",
            "type" : "string",
            "multiValued" : false,
            "description" : "A String that represents the subjects
              specified correct response to the corresponding
```

```
            challenge.",
          "required" : true,
          "caseExact" : false,
          "mutability" : "readWrite",
          "returned" : "default",
          "uniqueness" : "none"
        }
      ]
    },
    {
      "name" : "password",
      "type" : "string",
      "multiValued" : false,
      "description" : "A string value for the requested password.
      When specified, the challenges attribute must also be present.",
      "required" : true,
      "caseExact" : false,
      "mutability" : "writeOnly",
      "returned" : "never",
      "uniqueness" : "none"
    }
  ]
},
{
  "id" :
```



```
"urn:ietf:params:scim:schemas:core:2.0:password:PasswordValidateRequest",
    "name" : "Password Validate Request",
    "description" : "Used to submit a password for validation.",
    "attributes" : [
      {
        "name" : "password",
        "type" : "string",
        "multiValued" : false,
        "description" : "A string value for the requested password.
        When specified, the challenges attribute must also be present.",
        "required" : true,
        "caseExact" : false,
        "mutability" : "writeOnly",
        "returned" : "never",
        "uniqueness" : "none"
      }
    ]
  },
  {
    "id" :
"urn:ietf:params:scim:schemas:core:2.0:password:UserNameValidateRequest",
    "name" : "UserName Validate Request",
    "description" : "Used to submit a username for validation.",
    "attributes" : [
      {
        "name" : "$ref",
        "type" : "reference",
        "referenceTypes" : [
          "User"
        ],
        "multiValued" : false,
        "description" : "A reference value that contains a URI that
```

```
        points to the resource (e.g.  User) against which the proposed
        userName value is to be validated as an acceptable.",
        "required" : false,
        "caseExact" : false,
        "mutability" : "readWrite",
        "returned" : "default",
        "uniqueness" : "none"
      },
      {
        "name" : "userName",
        "type" : "string",
        "multiValued" : false,
        "description" : "A string value containing the requested userName
        value for which validation is requested.",
        "required" : true,
        "caseExact" : false,
```



```
        "mutability" : "readWrite",
        "returned" : "default",
        "uniqueness" : "none"
      }
    ]
  },
  {
    "id" :
"urn:ietf:params:scim:schemas:core:2.0:password:UserNameGenerateRequest",
    "name" : "Username Generate Request",
    "description" : "Used to request a new username be generated.",
    "attributes" : [
      {
        "name" : "$ref",
        "type" : "reference",
        "referenceTypes" : [
          "User"
        ],
        "multiValued" : false,
        "description" : "An reference value that contains a URI that
          points to the resource (e.g.  User) against which existing
          resource's name attribute MAY be used to generate a userName
          value.  When the $ref attribute is used, the generate
          request MUST be authenticated.",
        "required" : false,
        "caseExact" : false,
        "mutability" : "readWrite",
        "returned" : "default",
        "uniqueness" : "none"
      },
      {
        "name" : "userName",
        "type" : "string",
        "multiValued" : false,
        "description" : "A string value that is returned in the
          server's reponse that contains a generated userName value.
          The generated userName is not reserved and is guaranteed on
          first-come-first-served basis by a subsequent SCIM creation
          or modify request.",
        "required" : true,
        "caseExact" : false,
```

```
        "mutability" : "readOnly",
        "returned" : "default",
        "uniqueness" : "none"
      },
      {
        "name" : "name",
        "type" : "complex",
```



```
        "multiValued" : false,
        "description" : "An optional complex attribute containing the
          components of the user's name against which a userName value
          is to be generated. This attribute MAY be typically used as
          part of an anonymous userName generation request during a
          user registration dialog.",
        "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. Robert
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"
      }
    ]
  },
  {
    "id" :
"urn:ietf:params:scim:schemas:core:2.0:password:UserNameRecoverRequest",
    "name" : "UserName Recovery Request",
```



```
    "description" : "Used to look up a username by email address.",
    "attributes" : [
      {
```

```
      "name" : "email",
      "type" : "string",
      "multiValued" : false,
      "description" : "A string value containing an email address
        that is to be matched against an existing resource's emails
        attribue.",
      "required" : false,
      "caseExact" : false,
      "mutability" : "readWrite",
      "returned" : "default",
      "uniqueness" : "none"
    },
    {
      "name" : "userName",
      "type" : "string",
      "multiValued" : false,
      "description" : "A string value provided in response to a
        request which is the unique userName that corresponds to the
        recovery request.",
      "required" : true,
      "caseExact" : false,
      "mutability" : "readOnly",
      "returned" : "always",
      "uniqueness" : "none"
    }
  ]
 }
]
```

Request Schemas

4. Password Management ResourceTypes

The following are the resource type definitions for the resource types defined in this specification.

```
[
  {
    "schemas" : [
      "urn:ietf:params:scim:schemas:core:2.0:ResourceType"
    ],
    "id" : "PasswordPolicy",
    "name" : "Password Policy Definition",
    "endpoint" : "/Users",
    "description" : "Password policy definition",
```



```
    "schema" : "urn:ietf:params:scim:schemas:core:2.0:policy:Password",
    "schemaExtensions" : [

    ]
  },
  {
    "schemas" : [
      "urn:ietf:params:scim:schemas:core:2.0:ResourceType"
    ],
    "id" : "PasswordResetRequest",
    "name" : "Password Reset Request type",
    "endpoint" : "/PasswordResetRequest",
```

```
    "description" : "Resource type for processing password reset
      requests",
    "schema" :
"urn:ietf:params:scim:schemas:core:2.0:password:PasswordResetRequest",
    "schemaExtensions" : [

    ]
  },
  {
    "schemas" : [
      "urn:ietf:params:scim:schemas:core:2.0:ResourceType"
    ],
    "id" : "PasswordValidateRequest",
    "name" : "Password Validate Request type",
    "endpoint" : "/PasswordValidateRequest",
    "description" : "Resource type for processing password validation
      requests",
    "schema" :
"urn:ietf:params:scim:schemas:core:2.0:password:PasswordValidateRequest",
    "schemaExtensions" : [

    ]
  },
  {
    "schemas" : [
      "urn:ietf:params:scim:schemas:core:2.0:ResourceType"
    ],
    "id" : "UserNameValidateRequest",
    "name" : "Username Validate Request type",
    "endpoint" : "/UserNameValidateRequest",
    "description" : "Resource type for processing username validation
      requests",
    "schema" :
"urn:ietf:params:scim:schemas:core:2.0:password:UserNameValidateRequest",
    "schemaExtensions" : [
```



```
    ]
  },
  {
    "schemas" : [
      "urn:ietf:params:scim:schemas:core:2.0:ResourceType"
    ],
    "id" : "UserNameGenerateRequest",
    "name" : "Username Generation Request type",
    "endpoint" : "/UserNameGenerateRequest",
    "description" : "Resource type for processing username generation
      requests",
    "schema" :
"urn:ietf:params:scim:schemas:core:2.0:password:UserNameGenerateRequest",
    "schemaExtensions" : [

    ]
  },
  {
    "schemas" : [
      "urn:ietf:params:scim:schemas:core:2.0:ResourceType"
    ],
```

```
    "id" : "UserNameRecoverRequest",
    "name" : "Username Recovery Request type",
    "endpoint" : "/UserNameRecoverRequest",
    "description" : "Resource type for recovering usernames.",
    "schema" :
"urn:ietf:params:scim:schemas:core:2.0:password:UserNameRecoveryRequest",
    "schemaExtensions" : [

    ]
  }
]
```

Password Management Resource Types

5. Security Considerations

This specification builds on those of the SCIM API and Core-Schema specifications and as such the security considerations of both of these drafts apply to this specification.

[other considerations TBD]

6. IANA Considerations

TODO: Registration for Password management schema

TODO: Registration of password management resource types

7. References

7.1. Normative References

[I-D.ietf-scim-api]
Hunt, P., Grizzle, K., Ansari, M., Wahlstroem, E., and C. Mortimore, "System for Cross-Domain Identity Management: Protocol", draft-ietf-scim-api-14 (work in progress), December 2014.

[I-D.ietf-scim-core-schema]
Hunt, P., Grizzle, K., Wahlstroem, E., and C. Mortimore, "System for Cross-Domain Identity Management: Core Schema", draft-ietf-scim-core-schema-14 (work in progress), December 2014.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005.

7.2. Informative References

[I-D.ietf-precis-framework]
Saint-Andre, P. and M. Blanchet, "PRECIS Framework: Preparation, Enforcement, and Comparison of Internationalized Strings in Application Protocols", draft-ietf-precis-framework-21 (work in progress),

December 2014.

Appendix A. Contributors

Appendix B. Acknowledgments

The editor would like to thank the participants in the SCIM working group for their support of this specification.

Appendix C. Change Log

Draft 00 - PH - First Draft

Authors' Addresses

Phil Hunt (editor)
Oracle Corporation

Email: phil.hunt@yahoo.com

Gregg Wilson
Oracle Corporation

Email: gregg.wilson@oracle.com