



cutting through complexity

Saviynt APIs for User management from APIs(ServiceNow)

<Client>
KPMG Build

July 2022
Version: 1.0

Table of Contents

1	Document Control.....	3
1.1	Document Review & Feedback	3
1.2	Document Acceptance.....	3
1.3	Modification History.....	4
2	Document Purpose	5
3	Introduction.....	6
4	Saviynt Security Manager API Reference	6
4.1	Authentication	6
4.2	Create User	8
4.3	Update User	10
<CLIENT> Example	12	
4.4	Create Service Account.....	13
4.5	Create Privileged Account.....	18

1 Document Control

1.1 Document Review & Feedback

An updated version of this document has been created and will be reviewed by the team members and stakeholders listed below. The feedback obtained from their review will be incorporated.

Manjunath Madiraju Architect	07/20/2022 Date	Architect <Title>	07/20/2022 Date
Security Architect	Date	<Title>	Date
<Title>	Date	<Title>	Date

1.2 Document Acceptance

Representative Approvers

By signing this document, you confirm that you have read, reviewed, and approved the contents of this deliverable.

AGREED TO AND ACCEPTED BY:

Client Project Management

By: _____

Name: _____

Title: _____

Date: _____

AGREED TO AND ACCEPTED BY:

Vendor Project Management

By: _____

Name: _____

Title: _____

Date: _____

1.3 Modification History

Change Date	Author	Version Modified	Description of Changes
July 20, 2022	Manjunath Madiraju	1.0	Initial Creation

2 Document Purpose

Enabling access to technology resources in a secure and efficient manner is at the core of a strong cyber security program. An organization must provide its workforce (employees, contractors and business partners) with the required access to securely enable business operations and collaboration. The purpose of this document is to provide steps to create and update non-HR users from APIs, Manage users from all sources in Saviynt using APIs. These APIs can be invoked from ServiceNow and similar applications.

3 Introduction

Saviynt Security Manager provides REST APIs in order to perform Identity Administration that involves Create User, Update User, Get User Details and many other features.

<https://documenter.getpostman.com/view/1797923/RWaLwo21#intro>

4 Saviynt Security Manager API Reference

Syntax used in the API URLs:

{url}/ECM/{path}/apiName

example -

{url} - <https://example.saviyntcloud.com>

{path} - Use *api/v5* for SSM 5.2 or newer versions, Use *api* for older versions of SSM.

Note: If Content-Type in the response header needs to be overridden, add an entry in configuration table in database with name as 'api.v5.response.contenttype' and configdata as any valid content-type (application/json, application/xml etc.)

4.1 Authentication

POST Get Authorization Token

{url}/ECM/api/login

The `Bearer Authentication scheme` is dedicated to the authentication using a token. Even if this scheme comes from an OAuth2 specification, you can still use it as a way to exchange between a client and a server.

The SSM API currently requires the HTTP authentication scheme `Bearer` for authentication. All requests must have a valid `Token` specified in the HTTP Authorization header with the `Bearer` scheme.

{url}/ECM/api/login

A `POST` request by passing 'username' and 'password' in `json` text in the `body`, will return a `Token` in the response which can be passed in a `header` variable for subsequent requests

HEADERS

Content-Type – application/json

BODY raw

{"username":"admin","password":"password"}

<CLIENT> Example:

<https://<client>.saviyntcloud.com/ECM/api/login>

Content-Type – application/json

```
{"username":"saviynt-admin","password":"test-password"}
```

The screenshot shows the Postman application interface for testing an API endpoint. The URL is `https://[REDACTED].saviyntcloud.com/ECM/api/login`. The method is set to POST. The Body tab is selected, showing the JSON payload: `{"username": "mmadiraju", "password": "test-password"}`. The response status is 200 OK, with a response time of 202 ms and a size of 2.4 KB. The response body is displayed as JSON:

```
1 {  
2   "username": "mmadiraju",  
3   "roles": [  
4     "ROLE_ADMIN"  
5   ],  
6   "token_type": "Bearer",  
7   "access_token": "eyJhbGciOiJIUzI1NiJ9.
```

Response:

```
Ock5BenJVM1p5SGhwNG5YRkpDVUdxXBKbHR5VXdsZkp0N2NITHZUMzg2T3p6cHR5b0ExTW1UbTc4cDUxT0wwUFwvMjV2ZE1V
WFFRVzNnMFpMMkUxYnM1dVprb21WOXI5TXJuWDE5K1ByNzR1SFdMbEQxaTlmXC8zTWJzd2FLNjNwTEtjYVdiVjBJNklkcVwvcW40
bDg4V2J5cXkzMhdYIBjb0gwUjBtTHliVkVTVxh4cTFxSnE3NHQzSHUxRWEyOFhWaCtzZDd3cjNlempDVmNzNDRqNWZ0RmNMK3h
NRkswcjROZlI5OFTBIXC84Z2x1Y3dzc3VFUSTwOXZBUTFYTpHdlg5NIBEMz1ZWRCa1dMd115XC8rQXpEYUVBRVZBd0FBliwic3Viljoib
W1hZGlyYWp1liwiawF0ljoxNjM1OTY4NTI4LCJyb2xlcy16WyJST0xFX0FETUOII19.oBp8pn6ZD856Rt6smwrleRReBC18HEU2IOiibmne
Ww"
}
```

4.2 Create User

POST Create User

`{url}/ECM/{path}/createUser`

This API can be used to on-board or create a new user.

The `Authorization` must have `Bearer` followed by `Token`.

Optional Parameters:

`validateagainstpolicy` - Values: Y/N, default is Y. Checks for the password policy,

`statuskey` - Values: 1/0, 0 - Inactive, 1 - Active (default),

`allowpastdate` - true / false (default). If true, allows startdate to be less than current date,

`User params` like username, firstname, preferredFirstName, lastname, middlename, street, city, comments, statuskey, startdate(MM-dd-yyyy), enddate(MM-dd-yyyy), manager (manager username), password, location, jobCode, jobDescription, employeeType, departmentNumber, title, state, companyname, costcenter, departmentname, employeeclass, entity, jobcodedesc, locationdesc, locationnumber, siteid, orgunitid, region, regioncode, owner, employeeid, lastsyncdate, createdate, email, phonenumber, job_function, country, displayname, enabled(can be "1"/"true" or "0"/"false"), passwordExpired(can be "1"/"true" or "0"/"false"), accountExpired(can be "1"/"true" or "0"/"false"), accountLocked(can be "1"/"true" or "0"/"false"), secondaryManager, createdBy, termDate, vendorManager, secondaryPhone, secondaryEmail, customproperty<1-50>, hcp<1-5>, ecp<1-5>, customer(pass the organization name),

`securityQuestions` -

Example: "securityQuestions": [{ "securityQuestion": "What is your first Pet's name?", "securityAnswer": "kitty" }, { "securityQuestion": "What is your favorite food?", "securityAnswer": "food" }],

`checkrules` - true/ false (default : true), if true, then rules will be evaluated immediately, if false rules will be evaluated by a job. If `checkrulesforapi` configuration(true/false/null) is set in the configuration table, then it will take precedence over `checkrules` parameter.

`inlineruleevaluation` - true/ false (default is true), if true then rules will be evaluated immediately, if false rules will be evaluated by a job.

Note - CreateUser API will not invoke workflow based on the **User Modification Workflow** configuration. Please use `createUserRequest` API instead.

Note - Both `checkrules` and `inlineruleevaluation` need to be set as true for the user update rules to be evaluated immediately.

`batchidentifier` - When `inlineruleevaluation=false`, this Rule Run Unique Identifier can be set and user update rules can be processed later by calling `/api/processrules` API

It's highly recommended to pass `inlineruleevaluation:"false"` for bulk load and call `processrules` api to evaluate and run the rules after the load is done.

HEADERS

Content-Type – application/json

Authorization – Bearer {{token}}

BODY – raw

```
{
  "username": "johndoe",
  "firstname": "myfirstname",
  "startdate": "10-18-2018",
  "enddate": "10-11-2019",
  "statuskey": "1",
  "passwordExpired": "true",
  "enabled": "true",
  "accountExpired": "true",
  "accountLocked": "false",
  "allowpastdate": "true"
}
```

<CLIENT> Example:

{{url}}/ECM/{{path}}/createUser

<https://<client>.saviyntcloud.com/ECM/api/v5/createUser>

Content-Type – **application/json**

Authorization Type – **Bearer** (Token – Specify the **access_token** value we got from **4.1 Authorization** section.)

Body:

Create User (don't populate Username,systemusername,email – these will be auto generated)

```
{
  "firstname": "Rohit",
  "lastname": "Kulkarni",
  "middlename": "",
  "displayname": "Snow User",
  "preferredfirstname": "",
  "customproperty4": "10000350", // Manager's Employee_Number
  "statuskey": "1",
  "employeetype": "Contractor",
  "title": "Analyst",
  "street": "1, street Address",
  "city": "City",
  "state": "State",
  "country": "Country",
  "location": "LOC_009",
  "orgunitid": "Business_Unit",
  "costcenter": "Cost_Center",
  "companynamen": "Company",
  "departmentnamen": "Department_Name",
  "entity": "Division",
  "termdate": "",
  "startdate": "12-23-2021",
```

```
"customproperty1": "Zip123",
"customproperty20": "SERVICENOW",
"customproperty35": "true/false",
"customproperty36": "3rd party email"
}
```

https://[REDACTED].com/ECM/api/v5/createUser

POST https://[REDACTED].com/ECM/api/v5/createUser

Body (JSON)

```

1 {
2   "username": "rkulkarni3",
3   "firstname": "Rohit",
4   "lastname": "Kulkarni",
5   "startdate": "11-18-2021",
6   "enddate": "11-11-2022",
7   "statuskey": "1"
8 }
```

Body Cookies (6) Headers (16) Test Results

200 OK 582 ms 748 B Save Response

Pretty Raw Preview Visualize JSON

```

1 {
2   "errorCode": "0",
3   "message": "Users created with username rkulkarni3"
4 }
```

4.3 Update User

POST Update User

`{url}/ECM/{path}/updateUser`

This method updates a specific "user" record in SSM, based on the input parameter `username` of the relevant user.

All the input parameters it requires are attributes which could be updated for the relevant user record. The attributes which are supplied with valid values as input to the method, will get updated, if the operation is successfully executed.

The `Authorization` must have `Bearer` followed by `Token`.

Mandatory params:

`username`

OR

`propertytosearch`

Note - If both params are passed, `propertytosearch` will take precedence.

Optional params:

`validateagainstpolicy` - Values: Y/N, default is Y. Checks for the password policy,

`allowpastdate` - true / false (default). If true, allows startdate to be less than current date,

`statuskey` - Values: 1/0, 0 - Inactive, 1 - Active,

`updatedusername` - update username,

`User params` like firstname, preferredFirstName, lastname, middlename, street, city, comments, statuskey, startdate(MM-dd-yyyy), enddate(MM-dd-yyyy), manager (manager username), password, location, jobCode, jobDescription, employeeType, systemUserName, departmentNumber, title, state, companyname, costcenter, departmentname, employeeclass, entity, jobcodedesc, locationdesc, locationnumber, siteid, orgunitid, region, regioncode, owner, employeeid, lastsyncdate, createdate, email, phonenumbers, job_function, country, displayname, enabled(can be "1"/"true" or "0"/"false"), passwordExpired(can be "1"/"true" or "0"/"false"), accountExpired(can be "1"/"true" or "0"/"false"), accountLocked(can be "1"/"true" or "0"/"false"), secondaryManager, createdBy, termDate, vendorManager, secondaryPhone, secondaryEmail, customproperty<1-50>, hcp<1-5>, ecp<1-5>, customer(pass the organization name),

`securityQuestions` -

Example: "securityQuestions": [{ "securityQuestion": "What is your first Pet's name?", "securityAnswer": "kitty" }, { "securityQuestion": "What is your favorite food?", "securityAnswer": "food" }],

`checkrules` - true/ false (default : true), if true, then rules will be evaluated immediately, if false rules will be evaluated by a job. If checkrulesforapi configuration(true/false/null) is set in the configuration table, then it will take precedence over checkrules parameter.

`inlineruleevaluation` - true/ false (default is true), if true then rules will be evaluated immediately, if false rules will be evaluated by a job.

Note - UpdateUser API will not invoke workflow based on the **User Modification Workflow** configuration. Please use updateUserRequest API instead.

Note - Both `checkrules` and `inlineruleevaluation` need to be set as true for the user update rules to be evaluated immediately.

`batchidentifier` - When inlineruleevaluation=false, this Rule Run Unique Identifier can be set and user update rules can be processed later by calling /api/processrules API

Note - This API supports PUT HTTP method along with POST.

It's highly recommended to pass inlineruleevaluation:"false" for bulk load and call processrules api to evaluate and run the rules after the load is done.

HEADERS

Authorization – Bearer {{token}}

Content-Type – application/json

```
{
  "propertytosearch": "employeeid",
  "employeeid": "NW10010687",
  "firstname": "John",
  "preferredFirstName": "RoboID111111",
  "lastname": "Chipchase",
  "middlename": "kumar",
  "email": "test@sav.com",
```

```

    "employeeType": "RobotID11111",
    "jobcodedesc": "QA",
    "manager": "admin",
    "statuskey": "1",
    "customproperty1" : "customized value"
}

```

<CLIENT> Example

`{ {url} }/ECM/{ {path} }/updateUser`

<https://<client>.saviyntcloud.com/ECM/api/v5/updateUser>

Content-Type – **application/json**

Authorization Type – **Bearer** (Token – Specify the **access_token** value we got from **4.1 Authorization** section.)

Body:

Update User (**username** (saviynt attribute) is required in this update payload)

```

{
  "propertytosearch": "employeeid",
  "employeeid": "NW10010687", // Employee Number of user that should be updated.
  "firstname": "Rohit",
  "lastname": "Kulkarni",
  "middlename": "kumar",
  "displayname": "Snow User",
  "preferedfirstname": "",
  "customproperty4": "10000350", // Manager's Employee_Number
  "statuskey": "1",
  "employeeType": "Contractor",
  "title": "Analyst",
  "street": "1, street Address",
  "city": "City",
  "state": "State",
  "country": "Country",
  "location": "LOC_009",
  "orgunitid": "Business_Unit",
  "costcenter": "Cost_Center",
  "companyname": "Company",
  "departmentname": "Department_Name",
  "entity": "Division",
  "termdate": "",
  "startdate": "12-23-2021",
  "customproperty1": "Zip123",
  "customproperty20": "SERVICENOW",
  "customproperty35": "true/false"
}

```

The screenshot shows a Postman interface with the following details:

- Method:** POST
- URL:** https://<client>.saviyntcloud.com/ECM/api/v5/updateUser
- Body (JSON):**

```

1 {
2   ...
3   "propertytosearch": "lastname",
4   ...
5   "firstname": "Rohit",
6   ...
7   "preferredFirstName": "Rohit",
8   ...
9   "lastname": "Kulkarni",
10  ...
11  "middlename": "kumar",
12  ...
13  "email": "<REDACTED>",
14  ...
15  "employeetype": "Contractor",
16  ...
17  "jobcodedesc": "QA",
18  ...
19  "statuskey": "1"

```
- Response Status:** 200 OK
- Response Time:** 569 ms
- Response Size:** 734 B
- Response Body (Pretty JSON):**

```

1 {
2   ...
3   "errorCode": "0",
4   ...
5   "message": "User Updated Successfully"

```

4.4 Create Service Account

Step-1: Get Saviynt Username using EmployeeID

POST

Get User Details

`{url}/ECM/{path}/getUser`

This method returns a `List` of Users in SSM.

The `Authorization` must have `Bearer` followed by `Token`.

HEADERS

Content-Type

application/json

Authorization

Bearer `{token}`

<CLIENT> Payload:

URL: <https://<client>.saviyntcloud.com/ECM/api/v5/getUser>

Body: employeeid is the Employee Number of the user.

```
{
  "filtercriteria": {"employeeid": "111104802"},
  "responsefields": ["username"]
```

}

Response:

```
{
  "msg": "Successful",
  "displaycount": "1",
  "userlist": [
    {
      "userKey": 7430,
      "username": "10000000"
    }
  ],
  "totalcount": "1",
  "errorCode": "0"
}
```

[https://\[REDACTED\].vinyntcloud.com/ECM/api/v5/getUser](https://[REDACTED].vinyntcloud.com/ECM/api/v5/getUser)

POST https://[REDACTED].vinyntcloud.com/ECM/api/v5/getUser

Params Authorization Headers (11) Body Pre-request Script Tests Settings

Body (raw JSON)

```
1 {
2   "filtercriteria": {"employeeid": "[REDACTED]2"}, 
3   "responsefields": ["username"]
4 }
```

Body Cookies (6) Headers (16) Test Results Status: 200 OK

Pretty Raw Preview Visualize JSON

```
1 {
2   "msg": "Successful",
3   "displaycount": "1",
4   "userlist": [
5     {
6       "userKey": 7430,
7       "username": "10000000"
8     }
9   ],
10  "totalcount": "1",
```

Step-2: Check if Service Account name is unique**POST****Get Account Details**

{{url}}/ECM/{{path}}/getAccounts

This method returns a `List` of "Accounts" in SSM.The `Authorization` must have `Bearer` followed by `Token`.

Optional params: `username`, `endpoint`, `max`, `offset`, `accountQuery`, `advsearchcriteria` - accountKey,description,comments,accounttype,status,customproperty1-customproperty56,accountID,displayName,name,creator (username),updateuser (username),validfromDate

validthrough,createdon,lastlogondate,lastpasswordchange,updatedate,orphan (true/false), accountowner - with fields 1. type - user/ usergroup (mandatory), 2. value - username/ usergroup name (mandatory), 3. rank - 1 to 5 (optional)

Sample - "advsearchcriteria":{ "status":"ACTIVE","name":"john*","createdon":"2016-12-13","customproperty12": "*Ro*","orphan": "true","accountowner": [{ "type": "user", "value": "janedoe", "rank": "1" }]},

HEADERS

Content-Type

application/json

Authorization

Bearer {{token}}

<CLIENT> Payload:

URL: <https://<client>.saviyntcloud.com/ECM/api/v5/getAccounts>

Body: name is the name of the ServiceAccount.

```
{
  "accounttype": "Service Account",
  "endpoint": "Active Directory",
  "advsearchcriteria": {
    "name": "Saviynt1.sv"
  }
}
```

Response:

```
{
  "msg": "Accounts not found",
  "errorCode": "1"
}
```

The screenshot shows a Postman request configuration. The method is POST, the URL is <https://<client>.saviyntcloud.com/ECM/api/v5/getAccounts>, and the body is a JSON object:

```
{
  "accounttype": "Service Account",
  "endpoint": "Active Directory",
  "advsearchcriteria": {
    "name": "Saviynt1.sv"
  }
}
```

The response tab shows the following JSON output:

```

1
2   "msg": "Accounts not found",
3   "errorCode": "1"
4

```

Status: 200 OK

If there's a conflict with name field (account name), we need to increment by a number and invoke the getAccounts API again.

The screenshot shows a Postman interface with the following details:

- Method:** POST
- URL:** https://[REDACTED].avilyntcloud.com/ECM/api/v5/getAccounts
- Headers:** Authorization (set to Bearer {{token}}), Content-Type (set to application/json)
- Body:** JSON (selected)


```

1 {
2   ...
3   "accounttype": "Service Account",
4   ...
5   "endpoint": "Active Directory",
6   ...
7   "adsearcriteria": [
8     ...
9     {"name": "Saviynt.sv"}
10    ...
11  ]
12 }
      
```
- Status:** 200 OK
- Response Body (Pretty JSON):**

```

1 {
2   ...
3   "msg": "Successful",
4   ...
5   "displaycount": 1,
6   ...
7   "total": 1,
8   ...
9   "Accountdetails": [
10     {
11       ...
12     }
13   ]
14 }
      
```

Step-3: Invoke a createrequest API to create new Service Account with the username and the name (account name) values identified from steps-1 and 2 above.

POST

Request to Create New Service Account

`{{url}}/ECM/{{path}}/createrequest`

This API creates request for a new service account and assign it to user in SSM.

The `Authorization` must have `Bearer` followed by `Token`.

Mandatory params: `requesttype` - NEW, `username`, `endpoint`, `accountType`, `entitlement` -

If using older versions of SSM -

`entitlementType=entitlementValue`

If using SSM 5.2 or newer -

`"entitlement": [{"entitlementtype": "---", "entitlementvalue": "---", "startdate": "MM-dd-yyyy", "enddate": "MM-dd-yyyy", "businessjustification": "---"}]` (`startdate`, `enddate` and `businessjustification` are Optional params)

Optional params:

`accountname`, `securitysystem`, `comments`, `requestor`,

`owner` -

1. `ownerType` - mandatory field, values can be USER/USERGROUP,
2. `name` - mandatory field,
3. `rank` - optional field, values can be from 1 to 5.

HEADERS

Authorization

Bearer {{token}}

Content-Type

application/json

<CLIENT> Payload:

URL: <https://<client>.saviyntcloud.com/ECM/api/v5/createrequest>

Body :

```
{  
    "username": "10000010",  
    "requesttype": "NEW",  
    "endpoint": "Active Directory",  
    "securitysystem": "Active Directory",  
    "comments": "testser2acct",  
    "accountname": "<CLIENT>Serviceaccount2.sv",  
    "requestor": "admin",  
    "displayName": "Saviynt-test",  
    "customproperty20": "<CLIENT>Serviceaccount2@preprod.net",  
    "accountType": "Service Account",  
    "owner" : [  
        {  
            "ownerType" : "User",  
            "name" : "10000010",  
            "rank":1  
        }  
    ]  
}
```

Where

Username – value we got from step-1

requesttype: static value "NEW"

endpoint: static value "Active Directory"

securitysystem: static value "Active Directory"

comments: custom value "testser2acct"

accountname: unique account name from step-2 "<CLIENT>Serviceaccount2.sv"

requestor: static value "admin"

displayName: custom value "Saviynt-test"

customproperty20: dynamic value using accountName from step-2 @ domain name

accountType: static value "Service Account"

owner name : value we got from step-1

The screenshot shows a Postman interface with the following details:

- Method:** POST
- URL:** https://<client>.saviyntcloud.com/ECM/api/v5/createrequest
- Body:** JSON (selected)
- Request Body (Pretty):**

```

1 {
2   "username": "10000010",
3   "requesttype": "NEW",
4   "endpoint": "Active Directory",
5   "securitysystem": "Active Directory",
6   "comments": "testserver2account",
7   "accountname": "Saviynt-test",
8   "requestor": "admin",
9   "displayName": "Saviynt-test",
10  "customproperty7": "Saviynt",
11  "customproperty20": "Saviynt",
12  "accountType": "Service Account",
13  "owner": [
14    {
15      "ownerType": "User",
16      "name": "10000010",
17      "rank": 1
18    }
  
```
- Response Body (Pretty):**

```

1 {
2   "RequestId": "10717",
3   "requestkey": "44",
4   "errorCode": "0"
5 }
```
- Status:** 200 OK

4.5 Create Privileged Account

Step-1: Get Saviynt Username using EmployeeID

POST

Get User Details

{url}/ECM/{path}/getUser

This method returns a **List** of Users in SSM.

The **Authorization** must have **Bearer** followed by **Token**.

HEADERS

Content-Type

application/json

Authorization

Bearer {{token}}

<CLIENT> Payload:

URL: <https://<client>.saviyntcloud.com/ECM/api/v5/getUser>

Body: employeeid is the Employee Number of the user.

```
{
  "filtercriteria": {"employeeid": "111104802"},
  "responsefields": ["username"]
}
```

Response:

```
{
```

```

    "msg": "Successful",
    "displaycount": "1",
    "userlist": [
        {
            "userKey": 7430,
            "username": "10000000"
        }
    ],
    "totalcount": "1",
    "errorCode": "0"
}

```

https://[REDACTED]ECM/api/v5/getUser

POST https://[REDACTED]ECM/api/v5/getUser

Params Authorization Headers (11) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```

1  {
2    "filtercriteria": {"employeeid": "111104802"},
3    "responsefields": ["username"]
4  }

```

Body Cookies (6) Headers (16) Test Results Status: 200 OK

Pretty Raw Preview Visualize JSON

```

1  {}
2    "msg": "Successful",
3    "displaycount": "1",
4    "userlist": [
5        {
6            "userKey": 7430,
7            "username": "10000000"
8        }
9    ],
10   "totalcount": "1",

```

Step-2: Check if Privileged Account name is unique

POST

Get Account Details

`{url}/ECM/{path}/getAccounts`

This method returns a `List` of "Accounts" in SSM.

The `Authorization` must have `Bearer` followed by `Token`.

Optional params: `username`, `endpoint`, `max`, `offset`, `accountQuery`, `advsearchcriteria` - `accountKey`, `description`, `comments`, `accounttype`, `status`, `customproperty1`-
`customproperty56`, `accountID`, `displayName`, `name`, `creator` (`username`), `updateuser` (`username`), `validfromDate`
`validthrough`, `createdon`, `lastlogondate`, `lastpasswordchange`, `updatedate`, `orphan` (`true/false`),

accountowner - with fields 1. type - user/ usergroup (mandatory), 2. value - username/ usergroup name (mandatory), 3. rank - 1 to 5 (optional)

Sample - "advsearchcriteria":{ "status":"ACTIVE","name":"john*","createdon":"2016-12-13","customproperty12": "*Ro*","orphan": "true", "accountowner": [{ "type": "user", "value": "janedoe", "rank": "1" }]},

HEADERS

Content-Type

application/json

Authorization

Bearer {{token}}

<CLIENT> Payload:

URL: <https://<client>.saviyntcloud.com/ECM/api/v5/getAccounts>

Body: name is the name of the ServiceAccount.

```
{
  "accounttype": "Admin",
  "endpoint": "Active Directory",
  "advsearchcriteria": {
    "name": "testadmin"
  }
}
```

Response:

```
{
  "msg": "Accounts not found",
  "errorCode": "1"
}
```

The screenshot shows a Postman interface with a POST request to <https://<client>.saviyntcloud.com/ECM/api/v5/getAccounts>. The request body is a JSON object:

```
{
  "accounttype": "Admin",
  "endpoint": "Active Directory",
  "advsearchcriteria": {
    "name": "testadmin"
  }
}
```

The response tab shows a 200 OK status with the following JSON data:

```
{
  "msg": "Accounts not found",
  "errorCode": "1"
}
```

If there's a conflict with name field (account name), we need to increment by a number and invoke the getAccounts API again.

```

1 {
2   "accounttype": "Admin",
3   "endpoint": "Active Directory",
4   "adsearchcriteria": [
5     {"name": "ma", "value": "m", "operator": "contains", "type": "string", "id": 1}
6   ]
7 }
  
```

Status: 200 OK

Step-3: Invoke a createrequest API to create new Service Account with the username and the name (account name) values identified from steps-1 and 2 above.

POST

Request to Create New Privileged Account

`{url}/ECM/{{path}}/createrequest`

This API creates request for a new service account and assign it to user in SSM.

The `Authorization` must have `Bearer` followed by `Token`.

Mandatory params: `requesttype` - NEW, `username`, `endpoint`, `accountType`, `entitlement` -

If using older versions of SSM -

`entitlementType=entitlementValue`

If using SSM 5.2 or newer -

`"entitlement": [{"entitlementtype": "---", "entitlementvalue": "---", "startdate": "MM-dd-yyyy", "enddate": "MM-dd-yyyy", "businessjustification": "---"}]` (`startdate`, `enddate` and `businessjustification` are Optional params)

Optional params:

`accountname`, `securitysystem`, `comments`, `requestor`,
`owner` -

1. `ownerType` - mandatory field, values can be USER/USERGROUP,
2. `name` - mandatory field,
3. `rank` - optional field, values can be from 1 to 5.

HEADERS

Authorization

`Bearer {{token}}`

Content-Type

`application/json`

<CLIENT> Payload:

URL: <https://<client>.saviyntcloud.com/ECM/api/v5/createrequest>

Body :

```
{
  "username": "10000010",
  "requesttype": "NEW",
  "endpoint": "Active Directory",
  "securitysystem": "Active Directory",
  "comments": "testadminacct",
  "accountname": "<CLIENT>Admin2",
  "requestor": "admin",
  "accountType": "Admin"
}
```

Where

Username – value we got from step-1

requesttype: static value "NEW"

endpoint: static value "Active Directory"

securitysystem: static value "Active Directory"

comments: custom value "testser2acct"

accountname: unique account name from step-2 "<CLIENT>TestServiceaccount2.sv"

requestor: static value "admin"

accountType: static value "Admin"

POST https://[REDACTED].saviyntcloud.com/ECM/api/v5/createrequest

Params Authorization Headers (11) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```

1
2   "username": "10000010",
3   "requesttype": "NEW",
4   "endpoint": "Active Directory",
5   "securitysystem": "Active Directory",
6   "comments": "te[REDACTED]t",
7   "accountname": "[REDACTED]a",
8   "requestor": "admin",
9   "accountType": "Admin"
10
11
12

```

Body Cookies (6) Headers (16) Test Results

Pretty Raw Preview Visualize JSON

```

1
2   "RequestId": "10768",
3   "requestkey": "45",
4   "errorCode": "0"
5

```

4.6 Change Password from ServiceNow

POST

Change User Password

`{{url}}/ECM/{{path}}/changePassword`

This method resets "password" of a "user" record in SSM. Considering the input parameters, the value of new password should be supplied as `Password` and the `Username` should correspond to the user whose password is being reset.

The `Authorization` must have `Bearer` followed by `Token`.

Mandatory params:

`username`

`password`

Optional params:

`changePasswordAssociatedAccounts` - Values: true/false, default value - true, if true it creates change password tasks else just updates the user password

`endpoint` - list of endpoints comma separated (when `changePasswordAssociatedAccounts` is true)

`validateagainstpolicy` - Values: Y/N, default is Y. Checks against the password policy

`updateUserPassword` - Default value - true, (when `changePasswordAssociatedAccounts` is true). If `updateUserPassword` is true, update user password too along with creating the task. If `updateUserPassword` is false, just create the changepassword task.

`setarstasksource` - Values: true/false, default is false. If true, it will set source column in arstasks table with 'changeOwnPasswordFromAPI'. When source is 'changeOwnPasswordFromAPI',pwdLastSet is not set to "0" in ADconnector.

HEADERS

`Authorization`

`Bearer {{token}}`

BODY urlencoded

`username`

bliu

`password`

Hello\$?33892@woRLd

`changePasswordAssociatedAccounts`

true

`endpoint`

Workday

`validateagainstpolicy`

Y

`updateUserPassword`

false

`setarstasksource`

true

Contact us

Manjunath Madiraju
Manager, Technical Architect
Email mmadiraju@kpmg.com

Ken Dunbar
Engagement Director
Email kbdunbar@kpmg.com

© 2021 [legal member firm name], a [jurisdiction] [legal structure] and a member firm of the KPMG network of independent member firms affiliated with KPMG International Cooperative ("KPMG International"), a Swiss entity. All rights reserved.

The information contained herein is of a general nature and is not intended to address the circumstances of any particular individual or entity. Although we endeavour to provide accurate and timely information, there can be no guarantee that such information is accurate as of the date it is received or that it will continue to be accurate in the future. No one should act on such information without appropriate professional advice after a thorough examination of the particular situation.

The KPMG name, logo and "cutting through complexity" are registered trademarks or trademarks of KPMG International Cooperative ("KPMG International").