



User Management Component 2.9.2

UMC Service Layer API Developer Manual

Contents

What is UMC Service Layer?	1
Common Response Information	2
Authentication	3
Users	4
Groups	5
Roles	6
UMC APIs Error Codes	7
Account Policies	8
Function Rights	9
Change Password	10
Authentication IISAPPPOOL identity - only for test	11
Set Identity Property	12

Guidelines

This manual contains notes of varying importance that should be read with care; i.e.:

Important:

Highlights key information on handling the product, the product itself or to a particular part of the documentation.

Note: Provides supplementary information regarding handling the product, the product itself or a specific part of the documentation.

Trademarks

All names identified by ® are registered trademarks of Siemens AG.

The remaining trademarks in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owner.

Disclaimer of Liability

We have reviewed the contents of this publication to ensure consistency with the hardware and software described. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions.

Security information

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, systems, machines and networks. In order to protect plants, systems, machines and networks against cyber threats, it is necessary to implement – and continuously maintain – a holistic, state-of-the-art industrial security concept. Siemens' products and solutions only form one element of such a concept.

Customer is responsible to prevent unauthorized access to its plants, systems, machines and networks. Systems, machines and components should only be connected to the enterprise network or the internet if and to the extent necessary and with appropriate security measures (e.g. use of firewalls and network segmentation) in place.

Additionally, Siemens' guidance on appropriate security measures should be taken into account. For more information about industrial security, please visit <https://www.siemens.com/industrialsecurity>.

Siemens' products and solutions undergo continuous development to make them more secure. Siemens strongly recommends to apply product updates as soon as available and to always use the latest product versions. Use of product versions that are no longer supported, and failure to apply latest updates may increase customer's exposure to cyber threats.

To stay informed about product updates, subscribe to the Siemens Industrial Security RSS Feed under <https://www.siemens.com/industrialsecurity>.

Contents

1 What is UMC Service Layer?	5
2 Common Response Information	6
3 Authentication	7
3.1 Login	7
3.2 Authentication from Claim.....	9
3.3 Identity	9
4 Users	13
4.1 Get All Users	13
4.2 Get User Details.....	15
4.3 Create User.....	20
4.4 Update User - Basic.....	22
4.5 Update User - Full.....	23
4.6 Unlock User	25
4.7 Delete Users	26
4.8 Browse AD Users.....	27
4.9 Import Users	29
4.10 Reset Password.....	30
4.11 Get All System Users	31
5 Groups	34
5.1 Get All Groups	34
5.2 Get Group Details	35
5.3 Create Group	37
5.4 Update Group - Basic	38
5.5 Update Group - Full	40
5.6 Delete Groups.....	41
5.7 Browse AD Groups	43
5.8 Import Groups from AD	43
6 Roles	45
6.1 Get All Roles	45
6.2 Get Role Details.....	47
6.3 Create Role.....	47
6.4 Update Role	48
6.5 Delete Roles	50
7 UMC APIs Error Codes	52
8 Account Policies	59
8.1 Read Account Policies	59
8.2 Set Account Policies	59

8.3 Restore Default Account Policies.....	60
9 Function Rights	62
9.1 List Function Rights	62
10 Change Password.....	63
11 Authentication IISAPPPOOL identity - only for test	64
12 Set Identity Property.....	65

1 What is UMC Service Layer?

UMC Service Layer (SL) is the UMC Open Development Kit (ODK) to develop an application that needs to manage authentication, UMC users, groups and roles. Refer to the *User Management Component Installation Manual* for more details on the UMC Service Layer configuration.

UMC SL is based on HTTP services that reach a broad range of clients, including browsers and mobile devices. It can be used to develop Web pages and is also a powerful platform for exposing APIs providing services and data. HTTP is simple, flexible and ubiquitous. Almost any platform that you can think of has an HTTP library.

The data exchange is based on JSON format. Any call to the UMC SL include the requested "method" in the uri or the HTTP request. The response format is composed of:

- a [common part](#) that includes operation and protocol information;
- a detailed part that is described in each API.

For a complete list of the exposed APIs see:

- [authentication APIs](#);
- [user management APIs](#);
- [group management APIs](#).

Any request except the ones necessary to login, needs authorization. Authorization session is created using one of the ["authentication" methods](#) and transmitted by the browser in the request header (cookie authentication).

2 Common Response Information

The answer for every response (except the service for WebSSO) has the following format.

Response Header Fields

Field	Value
status	<ul style="list-style-type: none">• 200 in case of success;• 4xx otherwise.
Content-Type	application/json

Response JSON Example

```
"Version": 0 //the version of the protocol
"operation": "usersresult" // identify the operation called by the API
"result": 0
```

Response JSON Description

Property	Type	Description
version	integer	Protocol version. Consider that can vary depending on the API. See each API section for the value.
operation	string	Represents the invoked operation. See each API section for the value.
result	integer	Returns the last error code (decimal format) returned by the UMC APIs invoked during the command execution. See UMC APIs Error Codes for more details.

3 Authentication

The following APIs are dedicated to authentication:

- [Login](#)
- [Authentication from Claim](#)
- [Identity](#)

3.1 Login

Performs user authentication providing a user name and password.

example URL	https://localhost/UMC/slwapi/login											
Request HTTP Verb	POST											
Request Content-Type	application/x-www-form-urlencoded											
Request Parameters	The following parameters must be provided. <table border="1"><thead><tr><th>Name</th><th>Type</th><th>Description</th></tr></thead><tbody><tr><td>user</td><td>string</td><td>User name.</td></tr><tr><td>password</td><td>string</td><td>User password.</td></tr></tbody></table>			Name	Type	Description	user	string	User name.	password	string	User password.
Name	Type	Description										
user	string	User name.										
password	string	User password.										
Example:	https://localhost/UMC/slwapi/login?user=myuser&password=mypsw											

Response JSON Example

```
{"version" : 0 ,  
 "operation" : "loginresult",  
 "result" : 0}
```

Response JSON Description

For the common properties see [Common Response Information](#).

Example: REST call

```

function CallLogin([string]$protocol,[string]$srv,[string]$user,[string]$pwd)
{
    write-host "Call login to service layer"
    write-host "Server:$srv"
    write-host "user:$user"

    $url = $protocol+"://"+$srv+"/UMC/slwapi/login"
    $uri = New-Object "System.Uri" "$url"
    $request = [System.Net.HttpWebRequest]::Create($uri)
    $pair = "user=$user&password=$pwd"
    $enc = [system.Text.Encoding]::UTF8
    $data = $enc.GetBytes($pair)
    $request.ContentType = "application/x-www-form-urlencoded"
    $request.cookiecontainer = New-Object System.Net.CookieContainer
    $request.Method = "POST"
    $stream= $request.GetRequestStream()
    $stream.Write($data,0,$data.Length)
    $stream.Flush()
    $stream.Close()
    $response = $request.GetResponse()
    if ($response.StatusCode -eq "OK")
    {
        $cookie = $response.Cookies[0]
        $resstream= $response.GetResponseStream()
        $streadreader = New-Object System.IO.StreamReader($resstream)
        $logindata = $streadreader.ReadToEnd()
        write-host $logindata
        write-host $cookie
        return $cookie
    }
    return $null
}

```

Input Parameters

The input parameters of this function are the following:

- the **http** or **https** protocol,
- the server to be used for the REST call,
- the user name and password for the login.

Return Values

- NULL if the login is not successful,
- the cookie if the login is successful.

The function also writes the result JSON shown in the example.

3.2 Authentication from Claim

 This method is provided only for test purpose. Compatibility is not warranted and may be removed in a next version.

Performs the authentication of a user providing a claim. The user and the signature that guarantee claim validity are given in input. This user must have the proper function right (**UM_ADMIN** or **UM_CLAIMAUTH**). The name and the password in input are the credentials of the user that guarantees the claim validity and NOT the one of the user requesting authentication.

example URL	https://localhost/UMC/slwapi/pswclaimlogin																				
Request HTTP Verb	POST																				
Request Content-Type	application/x-www-form-urlencoded																				
Request Parameters	The following parameters must be provided. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>name</td> <td>string</td> <td>Name of the guaranteeing user.</td> </tr> <tr> <td>password</td> <td>string</td> <td>Password of the guaranteeing user.</td> </tr> <tr> <td>claim</td> <td>string</td> <td>Claim of the authenticating user.</td> </tr> <tr> <td>signature</td> <td>string</td> <td>Claim signature.</td> </tr> <tr> <td>keyid</td> <td>string</td> <td>The identifier of the key used to sign the claim.</td> </tr> </tbody> </table>			Name	Type	Description	name	string	Name of the guaranteeing user.	password	string	Password of the guaranteeing user.	claim	string	Claim of the authenticating user.	signature	string	Claim signature.	keyid	string	The identifier of the key used to sign the claim.
Name	Type	Description																			
name	string	Name of the guaranteeing user.																			
password	string	Password of the guaranteeing user.																			
claim	string	Claim of the authenticating user.																			
signature	string	Claim signature.																			
keyid	string	The identifier of the key used to sign the claim.																			
Example: https://localhost/UMC/slwapi/pswclaimlogin?name=myuser&password=mypsw&claim=myclaim&signature=mysignature&keyid=mykeyid																					

3.3 Identity

Provides information about the authenticated user.

example URL	https://localhost/UMC/slwapi/identity
Request HTTP Verb	POST
Request Content-Type	application/json
Request Parameters	None

Response JSON Example

```
{
  "domaininfo": {
    "domainid": "{G3F6FEE1931D8D38EBC79793DFEDECD1132FFBECB}",
    "domainname": "umcdomain"
  },
  "domains_support": "",
  "issuer": "https://vm-vdip13-26.swqa.tst/IPSimatic-Logon/",
  "language": "en-US",
  "result": 0,
  "rights": {
    "can_modify": true,
    "can_read": true,
    "can_register": true,
    "can_resetpwd": true,
    "can_unlock": true,
    "can_viewlog": true,
    "is_admin": true
  },
  "session_id": "f75306a84fc4448a9747058d41252b42",
  "user": {
    "fullname": "myfullname",
    "username": "myuser"
  },
  "validity": "2017-09-25T14:02:53.1557317Z",
  "version": 190
}
```

Response JSON Description

For the common properties see [Common Response Information](#).

Example: REST call

```
function CallLogin([string]$protocol,[string]$srv,[string]$user,[string]$pwd)
{
    write-host "Call identity from service layer"
    write-host "Server:$srv"

    $url = $protocol+"://"+$srv+"/UMC/slwapiv/identity"
    $uri = New-Object "System.Uri" "$url"
    $request = [System.Net.HttpWebRequest]::Create($uri)
    $request.ContentType = "application/json"
    $request.cookiecontainer = New-Object System.Net.CookieContainer
    $request.Header.Add($cookie)
```

```

$request.Method = "POST"
$stream= $request.GetRequestStream()
$stream.Flush()
$stream.Close()
$response = $request.GetResponse()
 ($response.StatusCode -eq "OK" )
{
    $cookie = $response.Cookies[0]
    $resstream= $response.GetResponseStream()
    $streadreader = New-Object System.IO.StreamReader($resstream)
    $responedata = $streadreader.ReadToEnd()
    write-host $responedata
}
$null
}

```

Properties

Property	Type	Description
domains_support	integer	Yes when the provisioning service is configurator (on the machine which exposes the identity service), otherwise no.
issuer	string	The address of the identity provider, which is configured on the machine that exposes the identity .
language	string	<p>It is the user language and has the format <langcode>-<countrycode>, where</p> <ul style="list-style-type: none"> • langcode is the language code according to the ISO 639 standard; we accept both two-letter codes (ISO 639-1) and three-letter codes (ISO 639-2); • countrycode is the country code according to the ISO 3166 standard. <p>An example is en-GB.</p>
results	integer	See Common Response Information .
session_id	integer	The identifier of the current session.
validity	string	The validity of the current session.
version	integer	See Common Response Information .

Domaininfo Object

The Domaininfo object has the following properties:

Property	Type	Description
domainid	integer	The identifier of the UMC domain.

Property	Type	Description
domainname	string	The name of the UMC domain.

Rights Object

Each rights object has the following properties:

Property	Type	Description
can_modify	string	If set to true the user can modify the umc configuration. If set to false the user cannot make modifications.
can_read	string	If set to true the user can view the configuration. If set to false the user cannot access the configuration.
can_register	string	If set to true the user can register a machine as a client. If set to false the user cannot register a client.
can_resetpwd	string	If set to true the user can reset thier password. If set to false the user cannot reset their password.
can_unlock	string	If set to true the user can unlock locked users. If set to false the user cannot unlock locked users.
can_viewlog	string	If set to true the user can view the umc log. If set to false the user cannot view the umc log.
is_admin	string	If set to true the user is a umc administrator. If set to false the user is not configured as administrator.

User object

Each user object has the following properties:

Property	Type	Description
Fullscreen	string	The full name of the authenticated user.
Username	string	The username of the authenticated user.

Return Values

- NULL if the login is not successful,
- the cookie if the login is successful.

The function also writes the result JSON shown in the example.

4 Users

The following APIs are dedicated to user management:

- [Get All Users](#)
- [Get User Details](#)
- [Create User](#)
- [Update User - Basic](#)
- [Update User - Full](#)
- [Unlock User](#)
- [Delete Users](#)
- [Browse AD Users](#)
- [Import Users](#)

4.1 Get All Users

Returns data related to all UMC users.

example URL	https://localhost/UMC/slwapiv1/users
Request HTTP Verb	POST
Request Content-Type	application/json
Request Parameters	None.

Response JSON Example

```
{
    "version": 0,
    "operation": "usersresult",
    "result": 0,
    "users":
    [
        {"id": 2, "objver": 1, "name": "root", "fullname": "", "comment": "", "userflags": 8},
        {"id": 1073741852, "objver": 2, "name": "umdom1\\administrator",
         "fullname": "Administrator", "comment": "Built-in account for
administering the computer\\domain", "userflags": 24},
        {"id": 29, "objver": 1, "name": "Marco", "fullname": "Marco Rossi", "comment": "
this user can change the password", "userflags": 10},
        {"id": 30, "objver": 2, "name": "Luca", "fullname": "Luca Bianchi",
         "comment": "this user is locked", "userflags": 12},
```

```

    {"id": 31,"objver":2,"name": "Isa","fullname": "Isa Verdi","comment": "this user must change the password", "userflags":9},
        {"id":1073741851, "objver":2,"name":"VM-UMC-11\\Administrator",
    "fullname": "", "comment": "", "userflags":40}
    ]
}

```

Response JSON Description

For the common properties see [Common Response Information](#).

users property Array

Each user object has the following properties:

Property	Type	Description
id	integer	User identifier, it is a 32 bit number greater than zero where the 31st bit is used to store the information if the user is imported or not. If the 31st bit is equal to 1 the user is imported from Active Directory, 0 otherwise. The ID ranges are as follows: <ul style="list-style-type: none"> • 1-20: reserved for UMC built in users; • 21- 1073741844: UMC users created from scratch; • 1073741845 to 2147483647: Imported users.
objver	integer	This is an integer that is incremented at each user modification.
name	string	User name.
fullname	string	User full name.
comment	string	A note that can be added to the user (optional).

Property	Type	Description
userflags	integer	<p>It is a 16 bit number bit mask representing the following flags, the order is from the LSB (less significant bit) to the MSB (most significant bit) :</p> <ol style="list-style-type: none"> 1. User Must Change Password: it is equal to 1 if the user must change password, 0 otherwise. 2. User Can Change Password: it is equal to 1 if the user can change password, 0 otherwise. 3. User Locked: it is equal to 1 if the user is locked, 0 otherwise. 4. User Enabled: it is equal to 1 if the user can authenticate into UMC, 0 otherwise. 5. User imported from AD: it is equal to 1 if the user has been imported from Active Directory domain users, 0 otherwise. 6. User imported from Local: it is equal to 1 if the user has been imported from Windows local users, 0 otherwise. 7. User imported from group: it is equal to 1 if the user has been imported through imported Active Directory group, 0 otherwise. 8. User offline: it is equal to 1 if the user is offline, 0 otherwise. See Get User Details for additional information. 9. <i>Not used.</i> 10. <i>Not used.</i> 11. <i>Not used.</i> 12. <i>Not used.</i> 13. <i>Not used.</i> 14. <i>Not used.</i> 15. <i>Not used.</i> 16. <i>Not used.</i>

4.2 Get User Details

Returns all the details related to the user associated with the input id.

example URL	https://localhost/UMC/slwapi/users{/id}
Request HTTP Verb	POST
Request Content-Type	application/json
Request Parameters	None.

Response JSON Example

{

```
"version": 140,
"operation": "usersresult",
"result": 0,
"users": {
    "id": 25,
    "objver": 16,
    "name": "mydomain\myName",
        "fullname": "my fullname",
        "comment": "",
        "firstname": "myFirstname",
        "lastname": "myLastname",
        "initials": "MI",
        "language": "IGNORE_CULT",
        "datalanguage": "IGNORE_CULT",
        "phone": "",
        "mobile": "",
        "email1": "",
        "email2": "",
        "email3": "",
        "sid": "",
        "expirationdate": "never expire",
        "alertsbeforeexpirationdate": "no allert",
        "passwordexpirationdays": "password never expire",
        "alertbeforepasswordexpirationdays": "no allert",
        "autologoff": "no autologoff",
        "timebeforeautologoff": "no autologoff",
        "imported": 1,
        "enabled": 1,
        "locked": 1,
        "mustchange": 1,
    "offline": 1,
    "importedfromgroup": 0 ,
        "importedfromad": 1 ,
        "canchange": 1,
    "attributes": [
        {
            "name": "attr_1",
            "value": "val_1"
        },
        {
            "name": "attr_2",
            "value": "val_2"
        }
    ],
    "roles": [
        {
            "name": "role_1",
            "id": 21,
            "description": "role_1 "
        },
        {

```

```
        "name": "role_2",
        "id": 22,
        "description": "role_2 "
    },
    {
        "name": "role_3",
        "id": 23,
        "description": "role_3 "
    }
],
"groups": [
    {
        "name": "group1",
        "id": 1,
        "imported": 1,
        "objver": 60,
        "sid":"",
        "description": "descr_1 "
    },
    {
        "name": "group2",
        "id": 2,
        "imported": 1,
        "objver": 60,
        "sid":"",
        "description": "descr_2 "
    }
]
}
```

Response JSON Description

For the common properties see [Common Response Information](#).

users Object

The user object has the following properties:

Property	Type	Description
id	integer	User identifier, it is a 32 bit number greater than zero where the 31st bit is used to store the information if the user is imported or not. If the 31st bit is equal to 1 the user is imported from Active Directory, 0 otherwise. The ID ranges are as follows: <ul style="list-style-type: none"> • 1-20: reserved for UMC built in users; • 21- 1073741844: UMC users created from scratch; • 1073741845 to 2147483647: Imported users.
objver	integer	This is an integer that is incremented at each user modification.
name	string	User name.
fullname	string	User full name.
comment	string	A note that can be added to the user (optional).
firstname	string	User first name.
lastname	string	User last name.
initials	string	User initials.
language	string	It is the user language and has the format <langcode>-<countrycode>, where <ul style="list-style-type: none"> • langcode is the language code according to the ISO 639 standard; we accept both two-letter codes (ISO 639-1) and three-letter codes (ISO 639-2); • countrycode is the country code according to the ISO 3166 standard. An example is en-GB.
datalanguage	string	It is the language in which are displayed the user data, see above for the language.
phone	string	User phone number.
mobile	string	User mobile phone number.
email1	string	First user email address.
email2	string	Additional email address for the user.
email3	string	Additional email address for the user.
sid	string	User Security Identifier (SID). See Microsoft Documentation on Security Identifiers for more details.
expirationdate	string	The date in which the user expires.

Property	Type	Description
alertsbeforeexpirationdate	string	It is the number of days from which a warning appears to the user notifying him/her about the user expiration.
passwordexpirationdays	string	It is the number of days for which the password is valid.
alertbeforepasswordexpirationdays	string	It is the number of days from which a warning appears to the user notifying him/her about the password expiration.
autologoff	string	It is equal to TRUE if the user automatic logoff is enabled, FALSE otherwise.
timebeforeautologoff	string	It is the number of minutes that must elapse before a user is automatically logged off from the system (session-based).
imported	integer	It is equal to 1 if the user has been imported from Windows local users, 0 otherwise.
enabled	integer	It is equal to 1 if the user can authenticate into UMC, 0 otherwise.
locked	integer	It is equal to 1 if the user is locked, 0 otherwise.
mustchange	integer	It is equal to 1 if the user must change password, 0 otherwise.
offline	integer	<p>It is equal to 1 if the user is offline, 0 otherwise. When creating a user, you can flag it as <i>offline</i>. UMC provisioning service checks if the offline user exists in Active Directory:</p> <ul style="list-style-type: none"> • if the user is present, user data are synchronized and the user becomes online, • otherwise the user remains offline. <p>The user name of offline users must follow the AD pattern <domainName><ADUserName>. They do not have a UMC password, as they cannot authenticate until they become online.</p>
importedfromgroup	integer	It is equal to 1 if the user has been imported through imported Active Directory group, 0 otherwise.
importedfromad	integer	It is equal to 1 if the user has been imported from Active Directory domain users, 0 otherwise.
canchange	integer	It is equal to 1 if the user can change password, 0 otherwise.

attributes property Array

Each attribute object has the following properties:

Property	Type	Description
name	string	Attribute name.
value	string	Attribute value.

roles property Array

Each role object has the following properties:

Property	Type	Description
name	string	Role name.
id	string	Role identifier.
description	string	Role description.

groups property Array

Each group object has the following properties:

Property	Type	Description
name	string	Group name.
id	string	Group identifier.
imported	integer	It is equal to 1 if the group is imported, 0 otherwise.
objver	integer	This is an integer that is incremented at each user modification.
sid	NA	Group Security Identifier (SID). See Microsoft Documentation on Security Identifiers for more details
description	string	Group description.

4.3 Create User

Creates a new user. In case of success the object is returned including the assigned id. Note that the password specified for the user is not bound to password policies.

example URL	https://localhost/UMC/slwapiv1/users/add
Request HTTP Verb	POST
Request Content-Type	application/json
Request Parameters	None.

Request JSON Example

It is optional. It is mandatory only if parameters in the query string are not present.

If the user is offline, the value of the **offline** property assumes the value **true** that corresponds to 1. If the user is not offline, the **offline** property is not part of the JSON file. The meaning is the same as described in [Get User Details](#). Empty passwords are not allowed.

```
[{"name": "franz",
  "password": "secret",
  "fullname": "franz",
  "canchange": true,
  "mustchange": false,
  "enabled": true,
  "locked": false,
  "offline": true}]
```

Request JSON Description

For the description of the single user properties see [Get User Details](#).

Response JSON Example

```
{
  "version": 0,
  "operation": "useraddresult",
  "result": 0, users: [
    {
      "id": 102, //present only in case of success
      "result": 0,
      "name": "franz",
      "offline": true,
      "password": "secret",
      "fullname": "franz",
      "canchange": true,
      "mustchange": false,
      "enabled": true,
      "locked": false
    }
}
```

Response JSON Description

For the common properties see [Common Response Information](#). For the description of the single user properties see [Get User Details](#). See above for the management of the **offline** property.

4.4 Update User - Basic

Updates the following basic user properties:

- **fullname**
- **password** (empty passwords are not allowed)
- **enabled**
- **mustchange**
- **canchange**

Note that the password specified for the user is not bound to password policies.

example URL	https://localhost/UMC/slwapiv/users/updateinline
Request HTTP Verb	POST
Request Content-Type	application/json
Parameters	None.

Request JSON Example

```
[  
  { "id":10247,  
    "name":"myName",  
    "objver":19,  
    "fullname":"user new fullname",  
    "password":"secret",  
    "canchange":1,  
    "mustchange":0,  
    "enabled":1,  
    "locked":0 }  
]
```

Request JSON Description

For the description of the single user properties see [Get User Details](#).

Response JSON Example

```
{ "version" : 0,  
  "operation" : "userupdateresult",  
  "result" : 0,  
  "users": [
```

```
{
    "canchange": 1,
    "enabled": 1,
    "fullname": "user new fullname",
    "id": 10247,
    "locked": 0,
    "mustchange": 0,
    "name": "myName",
    "objver": 20,
    "password": "secret",
    "result": 0
}
]
```

Response JSON Description

For the common properties see [Common Response Information](#). For the description of the single user properties see [Get User Details](#).

4.5 Update User - Full

Performs a full update of a user. All the modifiable user properties can be updated.

example URL	https://localhost/UMC/slwapi/users/update
Request HTTP Verb	POST
Request Content-Type	application/json
Request Parameters	None.

Request JSON Example

```
[
{
    "id": 5037,
    "objver": 2,
    "name": "Alessio",
    "fullname": "ROSSI ALESSIO",
    "comment": "GOOD",
    "initials": "AR",
    "datalanguage": "en-GB",
    "language": "en-GB",
    "firstname": "Alessio",
    "lastname": "Rossi",
    "mobile": ""
}
```

```

        "phone":"",
        "email1":"",
        "email2":"",
        "email3":"",
        "enabled":1,
        "locked":0,
        "canchange":1,
        "mustchange":0,
        "alertbeforepasswordexpirationdays":"0",
        "passwordexpirationdays":"0",
        "override_lock_policy":0,
        "alertsbeforeexpirationdate":"0",
        "autologoff":"30",
        "timebeforeautologoff":"3",
        "expirationdate":"Sun, 29 Nov 2076 08:54:34 GMT",
        "attributes":[
            { "name":"Address",
              "value":"Trafalgar Square 1a"      }
        ],
        "groups": [ ],
        "roles":[
            {"name":"myrole",
             "id":21,
             "description":"myroledescription"}
        ]
    }
]

```

Request JSON Description

For the description of the single user properties see [Get User Details](#).

Response JSON Example

```
{
"version" : 0 ,
"operation" : "userupdateresult",
"result" : 0,
"users":[
    {"alertbeforepasswordexpirationdays":"0",
     "alertsbeforeexpirationdate":"0",
     "attributes":
        [   { "name":"Address",
              "value":"Trafalgar Square 1a"      }
        ],
     "autologoff":"30",
     "canchange":1,

```

```
"comment": "GOOD",
"datalanguage": "en-GB",
"email1": "",
"email2": "",
"email3": "",
"enabled": 1,
"expirationdate": "Sun, 29 Nov 2076 08:54:34 GMT",
"firstname": "Alessio",
"fullname": "ROSSI ALESSIO",
"groups": [],
"id": 5037,
"initials": "AR",
"language": "en-GB",
"lastname": "Rossi",
"locked": 0,
"mobile": "+393211233210",
"mustchange": 0,
"name": "Alessio",
"objver": 3,
"override_lock_policy": 0,
"password": null,
"passwordexpirationdays": "0",
"phone": "0101112223",
"result": 0,
"roles": [
    {"description": "myroledescription1",
     "id": 21,
     "name": "myrole"}
],
"timebeforeautologoff": "3"
]
```

Response JSON Description

For the common properties see [Common Response Information](#). For the description of the single user properties see [Get User Details](#).

4.6 Unlock User

Unlocks a user.

example URL	https://localhost/UMC/slwapi/users/unlock
Request HTTP Verb	POST
Request Content-Type	application/json
Parameters	None.

Request JSON Example

```
{  
    name=newuser  
}
```

Request JSON Description

For the description of the single user properties see [Get User Details](#).

Response JSON Example

```
{  
    "version" : 0 ,  
    "operation" : "userunlockresult",  
    "result" : 0 }
```

Response JSON Description

For the common properties see [Common Response Information](#).

4.7 Delete Users

Deletes a list of users.

example URL	https://localhost/UMC/slwapi/users/delete{/id}
Request HTTP Verb	POST
Request Content-Type	application/json
Request Parameters	None.

Request JSON Example

Optional. Mandatory only if id is not present.

```
[  
    {"id":5034 }]
```

]

Request JSON Description

For the description of the single user properties see [Get User Details](#).

Response JSON Example

```
{
  "deleted": [ {
    "id": 5034,
    "result": 0
  }],
  "operation": "deleteresponse",
  "result": 0,
  "version": 0}
```

Response JSON Description

For the common properties see [Common Response Information](#).

deleted property Array

Each deleted object has the following properties:

Property	Type	Description
id	integer	Identifier of the user that has been deleted.
result	N/A	See Common Response Information .

4.8 Browse AD Users

Browses the users in the UMC database that have been imported from Active Directory (AD).

example URL	https://localhost/UMC/slwapiv1/adusers{/s*}
Request HTTP Verb	POST
Request Content-Type	application/json

example URL	https://localhost/UMC/slwapiv1/adusers{/s*}
Request Parameters	The search string. Example: https://localhost/UMC/slwapiv1/adusers/user* The service finds all users that match the string user* in the username field.

Response JSON Example

```
{
  "version": 0,
  "operation": "adusersresult",
  "result": 0,
  "users": [
    {
      "comment": "my comment 1",
      "email1": "myUser1@myDomain.com",
      "firstname": "Firstname1",
      "fullname": "Fullname of user 1",
      "lastname": "Lastname1",
      "mobile": "",
      "phone": "",
      "sid": "S-1-5-21-1972147420-2157339849-452057256-3749",
      "username": "domain\\USER1"
    },
    {
      "comment": "my comment 2",
      "email1": "myUser2@myDomain.com",
      "firstname": "Firstname2",
      "fullname": "Fullname of user 2",
      "lastname": "Lastname2",
      "mobile": "",
      "phone": "",
      "sid": "S-1-5-21-1972147420-2157339849-452057256-3753",
      "username": "domain\\USER2"
    },
    {
      "comment": "my comment 3",
      "email1": "myUser3@myDomain.com",
      "firstname": "Firstname3",
      "fullname": "Fullname of user 3",
      "lastname": "Lastname3",
      "mobile": "",
      "phone": "",
      "sid": "S-1-5-21-1972147420-2157339849-452057256-4077",
      "username": "domain\\USER3"
    }
  ]
}
```

Response JSON Description

In case of success an array of user objects, including the assigned identifiers, is returned. For the common properties see [Common Response Information](#). For the description of the single user properties see [Get User Details](#).

4.9 Import Users

Imports Active Directory (AD) or Windows local users.

example URL	https://localhost/UMC/slwapiv/users/import
Request HTTP Verb	POST
Request Content-Type	application/json
Request Parameters	None.

Request JSON Example Version 190 valid from UMC 1.9.1

```
[ {  
    "comment": "my comment",  
    "company": "my company",  
    "email1": "myMail@siemens.com",  
    "firstname": "username_first",  
    "fullname": "username_full",  
    "lastname": "username_last",  
    "localuser": true,  
    "mobile": "",  
    "phone": "",  
    "sid": "S-1-5-21-1972147420-2157339849-452057256-3753",  
    "username": "SWQA\\ITA00",  
    "selected": true,  
    "id": 1  
}]
```

Request JSON Example Version 0 valid prior to UMC 1.9.1

```
[ {  
    "comment": "my comment",  
    "company": "my company",  
    "email1": "myMail@siemens.com",  
    "firstname": "username_first",  
    "fullname": "username_full",  
    "lastname": "username_last",  
    "localuser": true,  
    "mobile": "",  
    "phone": "",  
    "sid": "S-1-5-21-1972147420-2157339849-452057256-3753",  
    "username": "SWQA\\ITA00",  
    "selected": true,  
    "id": 1  
}]
```

```
"lastname": "username_last",
"mobile": "",
"phone": "",
"sid": "S-1-5-21-1972147420-2157339849-452057256-3753",
"username": "SWQA\\ITA00",
"selected": true,
"id": 1
}]
```

Request JSON Description

For the description of the single user properties see [Get User Details](#).

Response JSON Example

```
{ }
```

4.10 Reset Password

Performs the password reset of a specified user. The new password must not necessarily conform to the Global Account Policy Properties. For more details on these properties, see *User Management Component API SDK Developer Manual*.

The user performing the action is represented by the input identity handle parameter and must have the function right **UM_ADMIN** or both the function rights **UM_VIEW** and **UM_RESETPSW**. For a detailed list of UM function rights, see the Appendix of *User Management Component API SDK Developer Manual*.

Empty passwords are not allowed.

example URL	https://localhost/UMC/slwapi/resetpsw
Request HTTP Verb	POST
Request Content-Type	application/json
Request Parameters	None.

Request JSON Example

```
[{"usertoreset" = "franz",
"pswtoreset" = "secret"}]
```

Request JSON Description

Property	Type	Description
usertoreset	string	Name of the user whose password has to be reset.
pswtoreset	string	The new value of the password.

Response JSON Example

```
{
    "version": 0 ,
    "operation" : "pswresetresult",
    "result" : 0
}
```

Response JSON Description

For the common properties see [Common Response Information](#).

4.11 Get All System Users

Returns data related to all imported windows local users and all imported virtual service account users.

For a description of this kind of users and how to import them into UMC see [Import Windows Local Users or Virtual User Accounts](#).

example URL	https://localhost/UMC/slwapiv1/systemusers
Request HTTP Verb	POST
Request Content-Type	application/json
Request Parameters	None.

Response JSON Example

```
{
    "version":0,
    "operation":"usersresult",
    "result":0,
    "users":
    [
        {
            "name": "Administrator"
        }
    ]
}
```

```

        {"id": 1073741852, "objver":2,"name":"localwks\\LOCAL SERVICE",
         "fullname":"LOCAL SERVICE USER","comment":"user comment","userflags":24},
        {"id": 1073741851, "objver":2,"name":"localwks\\$VUA$UMC SERVICE",
         "fullname":"","comment":"","userflags":24},
        {"id": 1073741853, "objver":2,"name":"localwks\\$IIS$umc_pool",
         "fullname":"","comment":"","userflags":24}

    ]
}

```

Response JSON Description

For the common properties see [Common Response Information](#).

users property Array

Each user object has the following properties:

Property	Type	Description
id	integer	User identifier, it is a 32 bit number greater than zero where the 31st bit is used to store the information if the user is imported or not. If the 31st bit is equal to 1 the user is imported from Active Directory, 0 otherwise. The ID ranges are as follows: <ul style="list-style-type: none"> • 1-20: reserved for UMC built in users; • 21- 1073741844: UMC users created from scratch; • 1073741845 to 2147483647: Imported users.
objver	integer	This is an integer that is incremented at each user modification.
name	string	User name.
fullname	string	User full name.
comment	string	A note that can be added to the user (optional).

Property	Type	Description
userflags	integer	<p>It is a 16 bit number bit mask representing the following flags, the order is from the LSB (less significant bit) to the MSB (most significant bit) :</p> <ol style="list-style-type: none"> 1. User Must Change Password: it is equal to 1 if the user must change password, 0 otherwise. 2. User Can Change Password: it is equal to 1 if the user can change password, 0 otherwise. 3. User Locked: it is equal to 1 if the user is locked, 0 otherwise. 4. User Enabled: it is equal to 1 if the user can authenticate into UMC, 0 otherwise. 5. User imported from AD: it is equal to 1 if the user has been imported from Active Directory domain users, 0 otherwise. 6. User imported from Local: it is equal to 1 if the user has been imported from Windows local users, 0 otherwise. 7. User imported from group: it is equal to 1 if the user has been imported through imported Active Directory group, 0 otherwise. 8. User offline: it is equal to 1 if the user is offline, 0 otherwise. See Get User Details for additional information. 9. <i>Not used.</i> 10. <i>Not used.</i> 11. <i>Not used.</i> 12. <i>Not used.</i> 13. <i>Not used.</i> 14. <i>Not used.</i> 15. <i>Not used.</i> 16. <i>Not used.</i>

5 Groups

The following APIs are dedicated to group management:

- [Get All Groups](#)
 - [Get Group Details](#)
 - [Create Group](#)
 - [Update Group - Basic](#)
 - [Update Group - Full](#)
 - [Delete Groups](#)
 - [Browse AD Groups](#)
 - [Import Groups from AD](#)

5.1 Get All Groups

Returns data related to all UMC groups.

example URL	https://localhost/UMC/slwapigroups
Request HTTP Verb	POST
Request Content-Type	application/json
Request Parameters	None.

Response JSON Example

```
{  
  "version" : 140 ,  
  "operation" : "groupsresult",  
  "result" : 0,  
  
  "groups" : [{  
    "description": "mygroupdescription",  
    "id": 628,  
    "imported": 0,  
    "name": "mygroup",  
    "objver": 0,  
    "offline": 0,  
    "sid": ""},  
  
  {  
    "description": "mygroupdescription1",  
    "id": 629,  
    "imported": 0,  
    "name": "mygroup1",  
    "objver": 0,  
    "offline": 0,  
    "sid": ""},  
  ]  
}
```

```

    "id":629,
    "imported":0,
    "name":"mygroup1",
    "objver":0,
    "offline":1,
    "sid":""} ]
}

```

Response JSON Description

For the common properties see [Common Response Information](#).

groups property Array

Each group object has the following properties:

Property	Type	Description
description	string	Group description.
id	integer	Group identifier.
imported	integer	It is equal to 1 if the group is imported from Active Directory, 0 otherwise.
name	string	Group name.
objver	integer	This is an integer that is incremented at each modification.
offline	integer	It is equal to 1 if the group is offline, 0 otherwise. When creating a group, you can flag it as <i>offline</i> . UMC provisioning service checks if the offline group exists in Active Directory: <ul style="list-style-type: none"> if the group is present, group data are synchronized, the AD users members of the groups are imported into UMC and the group becomes online, otherwise the group remains offline. The group name of offline users must follow the AD pattern <domainName>\<ADgroupName> .
sid	string	N/A This string is always empty.

5.2 Get Group Details

Returns all the details related to the group associated to the input id.

example URL	https://localhost/UMC/slwapi/groups/{/id}
Request HTTP Verb	POST
Request Content-Type	application/url-encoding

Request Parameters	None.
--------------------	-------

Response JSON Example

```
{
    "version" : 0 ,
    "operation" : "groupsresult",
    "result" : 0,
    "group" : {
        "name": "mygroup",
        "id" : 628,
        "imported" : 0,
        "offline" : 0,
        "objver" : 1,
        "description" : "mygroupnewdescription",
        "lastsync" : 0,
        "syncstatus" : "SYNC_NOSTATUS",
        "sadsstatus" : "SADS_SYNC",
        "sid" : "",
        "roles": [
            {
                "name": "myrole",
                "id" : 21,
                "description" : "myroledescription"
            }
        ],
        "users": []
    }
}
```

Response JSON Description

For the common properties see [Common Response Information](#).

Group Object

The group object has the following properties:

Property	Type	Description
name	string	Group name.
id	integer	Group identifier.
imported	integer	It is equal to 1 if the group is imported from Active Directory, 0 otherwise.

Property	Type	Description
offline	integer	<p>It is equal to 1 if the group is offline, 0 otherwise.</p> <p>When creating a group, you can flag it as <i>offline</i>. UMC provisioning service checks if the offline group exists in Active Directory:</p> <ul style="list-style-type: none"> If the group is present, group data are synchronized, the AD users members of the groups are imported into UMC and the group becomes online, Otherwise the group remains offline. <p>The group name of offline users must follow the AD pattern <domainName><ADgroupName>.</p>
objver	integer	This is an integer that is incremented at each modification.
description	string	Group description.
lastsync	N/A	<i>Not used.</i>
syncstatus	N/A	<i>Not used.</i>
sadsstatus	string	<p>SADS offline synchronization.</p> <p>Possible string values: "SADS_NOSTATUS" , "SADS_EMPTY", "SADS_SYNC"</p> <p>See Group SADS Syncronization Status Values for more details.</p>
sid	string	Group Security Identifier (SID). See Microsoft Documentation on Security Identifiers for more details.

roles property Array

Each role object has the following properties:

Property	Type	Description
name	string	Role name.
id	string	Role identifier.
description	string	Role description.

5.3 Create Group

Creates a new group. In case of success the object is returned including the assigned id.

example URL	https://localhost/UMC/slwapi/groups/add
Request HTTP Verb	POST
Request Content-Type	application/json
Request Parameters	None.

Request JSON Example

```
[  
  {"name":"mygroup",  
   "domain": "",  
   "description":"mygroupdescription",  
   "offline":true  
}]
```

Request JSON Description

For the description of the single group properties see [Get Group Details](#).

If the group is offline, the value of the **offline** property assumes the value **true** that corresponds to 1. If the group is not offline, the **offline** property is not part of the JSON file. The meaning is the same as described in [Get Group Details](#).

Response JSON Example

```
{  
  "groups": [  
    {"description":"mygroupdescription",  
     "domain": "",  
     "id":628,  
     "name":"mygroup",  
     "result":0},  
    {"operation":"useraddgroup",  
     "offline":true,  
     "result":0,  
     "version":0}
```

Response JSON Description

For the common properties see [Common Response Information](#). For the description of the single user properties see [Get Group Details](#). See above for the management of the **offline** property.

5.4 Update Group - Basic

Updates the following basic user properties:

- **name**,
- **domain**,

- **description.**

example URL	https://localhost/UMC/slwapi/groups/updateinline
Request HTTP Verb	POST
Request Content-Type	application/json
Request Parameters	None.

Request JSON Example

```
[{
  "id":628,
  "name":"mygroup",
  "domain":"UMC",
  "description":"mygroupnewdescription"
}]
```

Request JSON Description

For the description of the single group properties see [Get Group Details](#).

Response JSON Example

```
{
  "groups": [
    {
      "description": "mygroupnewdescription",
      "domain": "UMC",
      "id": 628,
      "name": "mygroup",
      "objver": 0,
      "result": 0
    }
  ],
  "operation": "userupdateinlinegroup",
  "result": 0,
  "version": 0
}
```

Response JSON Description

In case of success the object is returned including the assigned id. For the common properties see [Common Response Information](#). For the description of the single group properties see [Get Group Details](#).

5.5 Update Group - Full

Performs a full update of a group. All the modifiable group properties can be updated.

example URL	https://localhost/UMC/slwapiv/groups/update
Request HTTP Verb	POST
Request Content-Type	application/json
Request Parameters	None.

Request JSON Example

```
[ {
    "name": "mygroup",
    "id": 628,
    "imported": 0,
    "objver": 1,
    "offline": 0,
    "description": "mygroupnewdescription",
    "lastsync": 0,
    "syncstatus": "SYNC_NOSTATUS",
    "sid": "",
    "roles": [
        {
            "name": "myrole",
            "id": 21,
            "description": "myroledescription"
        }
    ],
    "users": [
        {
            "id": 5034,
            "name": "nuovo",
            "fullname": "nuovo"
        }
    ],
    "rights": {
        "can_modify": true,
        "can_read": true,
        "can_resetpwd": true,
        "can_unlock": true,
        "is_admin": true
    }
}]
```

Response JSON Description

For the description of the single user properties see [Get Group Details](#).

Response JSON Example

```
{
  "groups": [
    {
      "description": "mygroupnewdescription",
      "id": 628,
      "imported": 0,
      "lastsync": 0,
      "name": "mygroup",
      "objver": 1,
      "offline": 0,
      "result": 0,
      "rights": {
        "can_modify": true,
        "can_read": true,
        "can_resetpwd": true,
        "can_unlock": true,
        "is_admin": true
      },
      "roles": [
        {
          "description": "myroledescription",
          "id": 21,
          "name": "myrole"
        }
      ],
      "sid": "",
      "syncstatus": "SYNC_NOSTATUS",
      "users": [
        {
          "fullname": "nuovo",
          "id": 5034,
          "name": "nuovo"
        }
      ],
      "operation": "userupdategroup",
      "result": 0,
      "version": 0
    }
  ]
}
```

Response JSON Description

For the common properties see [Common Response Information](#). For the description of the single user properties see [Get Group Details](#).

5.6 Delete Groups

Deletes a list of groups.

example URL	https://localhost/UMC/slwapi/groups/delete
Request HTTP Verb	POST

Request Content-Type	application/json
Request Parameters	None.

Request JSON Example

```
[ {
  "id":628
}]
```

Request JSON Description

For the description of the single group properties see [Get Group Details](#).

Response JSON Example

```
{
  "groups": [ {
    "id":628,
    "result":0 },
    "operation":"deletegroupresponse",
    "result":0,
    "version":0 }
```

Response JSON Description

In case of success the object is returned including the assigned id. For the common properties see [Common Response Information](#).

groups property Array

Each groups object has the following properties:

Property	Type	Description
id	integer	Identifier of the group that has been deleted.
result	N/A	See Common Response Information .

5.7 Browse AD Groups

Browses the groups in the UMC database that have been imported from Active Directory (AD).

example URL	<code>https://localhost/UMC/slwapi/adgroups{/s*}</code>
Request HTTP Verb	POST
Request Content-Type	application/json
Request Parameters	<p>The search string. Example: <code>https://localhost/UMC/slwapi/adgroups/umc*</code> The service finds all groups that match the string umc* in the groupname field.</p>

Response JSON Example

```
{
  "version" : 0,
  "operation" : "adgroupsresult",
  "result" : 0,
  "groups": [
    {"description":"vdi_P13_UMC","domain":"MYDOMAIN","name":"vdi_P13_UMC",
     "sid":"S-1-5-21-1972147420-2157339849-452057256-14151"},
    {"description":"SWQA UMC","domain":"MYDOMAIN","name":"SWQA UMC","sid":
     "S-1-5-21-1972147420-2157339849-452057256-13147"},
    {"description":"SWQA UMC read","domain":"MYDOMAIN","name":"SWQA UMC
     read","sid":"S-1-5-21-1972147420-2157339849-452057256-15002"}
  ]
}
```

Response JSON Description

In case of success an array of groups objects, including the assigned identifiers, is returned. For the common properties see [Common Response Information](#). For the description of the single user properties see [Get Group Details](#).

5.8 Import Groups from AD

Imports groups from Active Directory (AD).

example URL	<code>https://localhost/UMC/slwapi/groups/import</code>
Request HTTP Verb	POST

Request Content-Type	application/json
Request Parameters	None.

Request JSON Example

```
[{"description":"SWQA UMC read",
 "domain":"SWQA",
 "name":"SWQA UMC read",
 "sid":"S-1-5-21-1972147420-2157339849-452057256-15002",
 "selected":true}]
```

Request JSON Description

For the description of the single user properties see [Get Group Details](#).

Response JSON Example

```
{"version" : 0,
 "operation" : "groupimportresult",
 "result" : 0,
 "groups": [
 {"description":"SWQA UMC read",
 "domain":"SWQA",
 "name":"SWQA UMC read",
 "result":13,
 "selected":true,
 "sid":"S-1-5-21-1972147420-2157339849-452057256-15002"
 } ]}
```

JSON Description

For the common properties see [Common Response Information](#). For the description of the single user properties see [Get Group Details](#).

6 Roles

The following APIs are dedicated to role management:

- [Get All Roles](#)
- [Get Role Details](#)
- [Create Role](#)
- [Update Role](#)
- [Delete Roles](#)

6.1 Get All Roles

Returns data related to all UMC roles.

example URL	https://localhost/UMC/slwapiv/roles
Request HTTP Verb	POST
Request Content-Type	application/json
Request Parameters	None.

Response JSON Example

```
{
    "Version": 0,
    "operation": "rolesresult",
    "result": 0,
    "roles":
    [
        {
            "id": 2, "name": "administrator", "description": "", "application": "UMC", "function_rights": "all" },
            {"id": 21, "name": "Reader", "description": "Only Read", "application": "UMC", "function_rights": []}
            {"id": 22, "name": "Writer", "description": "Write", "application": "UMC", "function_rights": []
                [
                    { "name": "UM_ADMIN", "id" : 10001, "description" : "Administer UM Configuration" },
                    { "name": "UM_VIEW", "id" : 10002, "description" : "View UM Configuration" },
                    { "name": "UM_RESETPWD", "id" : 10003, "description" : "Reset user password" },
                ]
            }
        ]
    ]
}
```

```

        { "name": "UM_RA", "id" : 10004, "description" : "Login from Remote Authentication" },
        { "name": "UM_UNLOCKUSR", "id" : 10005, "description" : "Unlock User" },
        { "name": "UM_JOIN", "id" : 10006, "description" : "Create UM Server"
    }
]
}
]
}

```

Response JSON Description

For the common properties see [Common Response Information](#).

roles property Array

Each role object has the following properties:

Property	Type	Description
id	integer	Role identifier. 0 and negative IDs are not allowed. The ID ranges are as follows: <ul style="list-style-type: none"> • 1-20: reserved to system roles; • 21-32600: UMC roles created from scratch.
name	string	Role name.
description	string	Role description.
application	-	<i>Not used.</i>
function_rights	function right Array	The set of function rights associated to the role. If the role has no associated function rights the value is [] , if the role has all the associated function rights the value is all , this is the case of the Administrator built-in role. See below for the description of the function right object properties.

function rights property Array

Each function right object has the following properties:

Property	Type	Description
name	string	Function right name.
id	integer	Function right identifier.
description	string	Function right description.

6.2 Get Role Details

Returns all the details related to the role associated with the input id.

example URL	https://localhost/UMC/slwapiv/roles/{/id}
Request HTTP Verb	POST
Request Content-Type	application/json
Request Parameters	None.

Response JSON Example

```
{
  "version" : 0 ,
  "operation" : "roledetailresult",
  "result" : 0,
  "role" :
  {
    "name": "myrole",
    "objver": 6,
    "id" : 21,
    "application" : "UMC",
    "description" : "myroledescription1a",
    "function_rights" : [ "name": "UM_VIEW", "id" : 10002, "description" :
      "View UM Configuration"]
  }
}
```

Request JSON Description

For the common properties see [Common Response Information](#), whereas for the description of the single role properties see [Get All Roles](#).

6.3 Create Role

Creates a new role. In case of success the object is returned including the assigned id. The number of roles present in the system cannot exceed 200. In addition a database constraint on the role identifiers exists. In case you get an error message that no more role identifiers are available, to create new roles, you have first to purge the existing one with the corresponding **umconf** command. See the *UMCONF User Manual* for more details.

example URL	https://localhost/UMC/slwapiv/roles/add
--------------------	---

Request HTTP Verb	POST
Request Content-Type	application/json
Request Parameters	None.

Request JSON Example

Optional. It is mandatory only if parameters in the query string are not present.

```
[{"name": "myrolename",
"description": "myroledescr"}]
```

Request JSON Description

For the description of the single role properties see [Get All Roles](#).

Response JSON Example

```
{
  "operation": "addroleresult",
  "roles": [
    {
      "description": "myroledescr",
      "id": 21,
      "name": "myrolename",
      "result": 0
    }
  ],
  "version": 0
}
```

Response JSON Description

For the common properties see [Common Response Information](#). For the description of the single user properties see [Get All Roles](#).

6.4 Update Role

Performs a full update of a role. All the modifiable role properties can be updated.

example URL	https://localhost/UMC/slwapi/roles/update
Request HTTP Verb	POST

Request Content-Type	application/json
Request Parameters	None.

Request JSON Example

```
[ {
    "name": "myrole",
    "objver": 2,
    "id": 21,
    "application": "UMC",
    "description": "myroledescription1",
    "function_rights": [{"name": "UM_ADMIN", "id": 10001, "description": "Administer UM Configuration"}],
    "rights": [{"name": "UM_ADMIN", "id": 10001, "description": "Administer UM Configuration", "own": true},
               {"name": "UM_UNLOCKUSR", "id": 10005, "description": "Unlock User", "own": true}]
}]
```

Request JSON Description

For the description of the single role properties see [Get All Roles](#).

Response JSON Example

```
{
  "operation": "roleupdate",
  "result": 0,
  "roles": [
    {
      "application": "UMC",
      "description": "myroledescription1",
      "function_rights": [{"name": "UM_ADMIN", "id": 10001, "description": "Administer UM Configuration"}],
      "id": 21,
      "name": "myrole",
      "objver": 2,
      "result": 0,
      "rights": [
        {
          "description": "Administer UM Configuration",
          "id": 10001,
          "name": "UM_ADMIN",
          "own": true}
        {
          "name": "UM_UNLOCKUSR",
```

```
        "id":10005,  
        "description":"Unlock User",  
        "own":true}  
    ],  
    "version":0  
}
```

Response JSON Description

For the common properties see [Common Response Information](#). For the description of the single role properties see [Get All Roles](#).

6.5 Delete Roles

Deletes a list of users.

example URL	https://localhost/UMC/slwapi/roles/delete{/id}
Request HTTP Verb	POST
Request Content-Type	application/json
Request Parameters	None.

Request JSON Example

Optional. Mandatory only if id is not present.

```
[  
  {"id":22}  
]
```

Request JSON Description

For the description of the single role properties see [Get All Roles](#).

Response JSON Example

```
{  
  "operation":"roledelete",
```

```
"result":0,  
"roles": [{  
  "id":22,  
  "result":0}],  
  "version":0  
}
```

Response JSON Description

For the common properties see [Common Response Information](#), for the description of the single role properties see [Get All Roles](#).

7 UMC APIs Error Codes

All the UMC APIs return a boolean value or an object handle. If the API is successful, the returned boolean value is true or the object handle is well formed; otherwise the returned boolean value is false, or null is returned instead of the object handle. If the API fails an error code can be retrieved calling the **SL_GetLastError** method. **SL_RESULT** defines the type of error. In what follows we list the possible error codes.

Generic Errors

Name	Hexadecimal Value	Decimal Value	Description
SL_SUCCESS	0X00	0	No errors have occurred.
SL_GENERROR	0X01	1	Generic error.
SL_BAD_HANDLE	0x114	276	Internal error for invalid handle.
SL_NOSESSION	0X30	48	The Web session is expired.

Authentication Errors

Name	Hexadecimal Value	Decimal Value	Description
SL_USERLOCKED	0X02	2	The user for whom you want to perform the authentication is locked.
SL_USERDISABLED	0X03	3	The user for whom you want to perform the authentication is disabled.
SL_WRONGUSERNAMEPASSWORD	0X04	4	During the authentication phase, the user name or password are incorrect.
SL_PASSWORDPOLICYVIOLATION	0X05	5	Password policy violation (determined by UMC account policies). For a detailed list of Account Policies, see <i>User Management Component API SDK Developer Manual</i> .
SL_USERMUSTCHANGEPASSWORD	0X06	6	The user password must be changed.
SL_PASSWORDEXPIRED	0X07	7	The user password is expired.
SL_FAILED	0X0A	10	Generic operation failed.

Name	Hexadecimal Value	Decimal Value	Description
SL_ALREADYLOCKED	0X0B	11	The UMC object is already locked.
SL_COMMERR	0X0C	12	Transmission/Communication error.
SL_NOTIMPL	0X10	16	Returned if a not implemented method is invoked.
SL_CHANGEPSWDISABLE	0X19	25	The user cannot change the password.
SL_USERUNKNOWN	0X20	32	The user is not present in the system.
SL_USERNEVEREXPIRE	0X21	33	The user never expires.
SL_TICKETEXPIRED	0X22	34	The authentication ticket is expired.
SL_USER_EXPIRED	0x27	39	The user is expired.
SL_PSWMINLEN_ERR	0x120	288	The account policy related to the minimal password length has been violated.
SL_PSW_CHANGE_FAIL	0X154	340	Password change failure.
SL_INVALID_NONCE	0x166	358	Login failed: invalid token. This event may occur if you try to access the login page directly from the URL or if you leave the login page open.
SL_WEAK_AUTH	0x167	359	Login failed: access not allowed using weak authentication method.

CRUD Operation Errors

Name	Hexadecimal Value	Decimal Value	Description
SL_ALREADYEXIST	0x0D	13	The UMC object already exists.
SL_LOCK_NEEDED	0x23	35	A lock is needed to complete the operation.
SL_NOT_LOCKED	0x24	36	The UMC object is not locked so you cannot unlock it.

Name	Hexadecimal Value	Decimal Value	Description
SL_OBJVERMISMATCH	0X31	49	A UMC object has been simultaneously modified by two Web UI instances and an object version mismatch has been detected.
SL_INVALID_OPERATION	0x103	259	The operation cannot be performed on the selected object.
SL_OBJ_DOES_NOT_EXIST	0x111	273	The UMC object does not exist or has not yet been saved into the UMC database.
SL_OBJECT_LOCKED_IN_DATABASE	0X153	339	The UMC object is already locked.
SL_FAIL_NOTAMASTER	0x160	352	An attempt has been made to modify the UMC database on a machine that is not a master.
SL_FAIL_BINDING_ADMIN_ROLE	0x161	353	An attempt has been made to assign the Administrator role to a group or the user who performed the association, either a UMX user or a Web UI user, does not have the Administrator role.
SL_OBJ_OFFLINE	0x0F	15	The user/group for which you want to perform an operation is offline and the operation is not allowed for offline objects.
SL_INVALID_NAME_FOR_OFFLINE_OBJ	0x165	357	The offline user/group that you are creating does not follow the pattern <domainName>\<objName>.
SL_INVALID_SID	0x5C	92	Invalid User Security Identifier (SID). See Microsoft Documentation on Security Identifiers for more details.

Provider Operation Errors

Name	Hexadecimal Value	Decimal Value	Description
SL_INVALID_PROVIDER	0x100	256	Operation not provided by this provider.
SL_INVALID_HANDLE	0x101	257	An invalid handle was passed as parameter.
SL_ERROR_LOADING_PROVIDER	0x102	258	An error occurred when loading the provider.

Internal or Parameter Errors

Name	Hexadecimal Value	Decimal Value	Description
SL_INVALID_PARAMETERS	0x104	261	The method has an incorrect parameter.
SL_MEMORY_ERROR	0x105	262	Memory allocation error.
SL_INITIALIZATION_ERROR	0x106	263	Initialization error.
SL_INVALID_LOCK_OPTION	0x108	264	The lock option has not been defined.
SL_INVALID_PROPERTY	0x109	265	The property has not been defined for the object.
SL_INVALID_CULTURE	0x17B	379	Invalid language

File Errors

Name	Hexadecimal Value	Decimal Value	Description
SL_ACCESS_FILE_ERROR	0x112	274	Access file error.
SL_UNKNOWN_FILE_FORMAT	0x113	275	Unknown file format.
SL_FILE_NOT_FOUND	0x50	80	File not found.
SL_PATH_NOT_FOUND	0x51	81	Path not found.
SL_FILE_CREATION_FAIL	0x52	82	Error during file creation.
SL_PATH_CREATION_FAIL	0x53	83	Error during path creation.
SL_INVALID_PATH	0x54	84	Invalid path.

Function Rights Errors

Name	Hexadecimal Value	Decimal Value	Description
SL_RESOURCE_NOT_FOUND	0x150	336	The user does not have the correct function right to perform the requested operation. This error has the same meaning as the SL_MISSING_FUNCTION_RIGHT error.
SL_INVALID_RESOURCE	0x151	337	The function right does not exist.
SL_MISSING_FUNCTION_RIGHT	0x152	338	The user does not have the correct function right to perform the requested operation. This error has the same meaning as the SL_RESOURCE_NOT_FOUND error.

Service Layer Errors

Name	Hexadecimal Value	Decimal Value	Description
SL_CLAIM_EXPIRED	0X155	341	The claim is expired.
SL_CLAIM_INVALID	0X156	342	The claim is invalid.
SL_JSON_ERROR	0X157	343	The .json file is not well formed.
SL_MKTKT_FAILURE	0X158	344	The "make ticket" operation failed.
SL_ABORTED	0x159	345	Operation aborted.

Package Errors

Name	Hexadecimal Value	Decimal Value	Description
SL_PACKAGE_CREATION_FAIL	0x55	85	Package creation failed.
SL_PACKAGE_COMPRESSION_FAIL	0x56	86	Package compression failed.
SL_PACKAGE_UNCOMPRESSION_FAIL	0x57	87	Package decompression failed.
SL_PACKAGE_ENCRYPTION_FAIL	0x58	88	Package encryption failed.
SL_PACKAGE_DECRYPTION_FAIL	0x59	89	Package decryption failed.

Name	Hexadecimal Value	Decimal Value	Description
SL_PACKAGE_RESTORE_FAIL	0x5A	90	Package restore failed.
SL_PACKAGE_WRONG_PASSWORD	0x5B	91	Wrong password for the package.

Database Errors

Name	Hexadecimal Value	Decimal Value	Description
SL_DBFILE_ACCESS_DENIED	0X32	50	The user cannot access a UMC database file.
SL_DBFILE_ERROR	0X33	51	Generic UMC database file error.
SL_DBFILE_OUT_OF_SPACE	0X34	52	A UMC database file is full.
SL_TOO_MANY_GROUPS	0X36	102	Too many groups assigned to a user.
SL_TOO_MANY_ROLES	0X37	103	Too many roles assigned to a user or group.
SL_TOO_MANY_USERS	0X38	104	Too many users assigned to a group.
SL_ROLEIDS_OUT_OF_SPACE	0X35	53	No more role IDs are available in the role database file. A purge of the roles is needed.

User Alias Errors

Name	Hexadecimal Value	Decimal Value	Description
SL_INVALID_USER_ALIAS	0x5E	94	Invalid user alias name.
SL_USER_ALIAS_ALREADY_EXIST	0x5F	95	User alias already exists.
SL_BAD_PKI_FILTER_NAME	0x115	277	Invalid filter name or filter name not present when authmode = SL_PKI_FILTER_MASK.

Secure Application Data Support (SADS) Errors

Name	Hexadecimal Value	Decimal Value	Description
SL_INVALID_DOMAIN_NAME	0x60	96	Invalid domain name.

Name	Hexadecimal Value	Decimal Value	Description
SL_NOT_CURRENT_DOMAIN	0x61	97	Input domain name is not the current domain.
SL_INVALID_KEY	0x70	112	Invalid key.
SL_KEY_GENERATION_FAIL	0x71	113	Error during key generation.
SL_KEY_ENCRYPTION_FAIL	0x72	114	Error during key encryption.
SL_KEY_DECRYPTION_FAIL	0x73	115	Error during key decryption.
SL_KEY_NOT_FOUND	0x74	116	Key not found.
SL_KEY_ENCRYPTION_NOT_ENABLED	0x75	117	Application key protection (global policies) not enabled.
SL_MAX_NUM_KEY	0x76	118	The maximum number of allowed keys has been reached.
SL_KEY_DECRYPTION_NO_ID_FOUND	0x77	119	No SUID of the identity has been found in EAK array.
SL_SADS_VERSION_ERROR	0x78	120	Wrong SADS version.
SL_WRONG_IDENTITY	0x79	121	Ticket authentication error while decrypting a key.
SL_EAK_BAD_FORMAT	0x80	128	Bad format of the encryption application object.
SL SUBJECT NOT ENABLED	0x81	129	Encryption not enabled for the specified subject.
SL SUBJECT KEY OBSOLETE	0x82	130	The decryption has been executed using an obsolete key.

8 Account Policies

8.1 Read Account Policies

command	getglobalaccountpolicies
example URL	https://localhost/UMC/slwapiv1/getglobalaccountpolicies
HTTP Verb	POST
Parameters	none
Data	
Result	<pre>{ "version" : 0 , "operation" : "getaccountpolicyanswer", "result" : 0, "global_account_policies":{ "enableLockAfterNAttempts":"1", "enablePasswordHistoryByDays":"1", "enablePasswordHistoryByNum":"0", "maxLoginErrors":"10", "minDaysBeforePwdReuse":"120", "numPwdBeforeReuse":"5", "passwordAging":"60", "passwordMaxLen":"120", "passwordMinAlphaChar":"2", "passwordMinLen":"8", "passwordMinLowChar":"1", "passwordMinNumericChar":"1", "passwordMinOtherChar":"0", "passwordMinUpChar":"1"} }</pre>
<hr/> Important: <hr/> <p>Status is 400 in case of error.</p> <hr/>	

8.2 Set Account Policies

command	setglobalaccountpolicies
----------------	--------------------------

8.3 Restore Default Account Policies

example URL	https://localhost/UMC/slwapi/setglobalaccountpolicies
HTTP Verb	POST
Parameters	none
Data	<pre>{ "passwordMinLen": "8", "passwordMaxLen": "120", "passwordMinLowChar": "1", "passwordMinUpChar": "1", "passwordMinAlphaChar": "2", "passwordMinNumericChar": "1" , "passwordMinOtherChar": "0", "passwordAging": "60", "enableLockAfterNAttempts": true, "maxLoginErrors": "11", "enablePasswordHistoryByDays": "1" , "minDaysBeforePwdReuse": "120", "enablePasswordHistoryByNum": "0", "numPwdBeforeReuse": "5", "passwordHistoryPolicy": "by-days" }</pre>
Result	<pre>{ "version" : 0 , "operation" : "setaccountpolicyanswer", "result" : 0 }</pre> <p>Important: Status is 400 in case of error. <u>"passwordHistoryPolicy": "by-num" in Data if "enablePasswordHistoryByNum": "1",</u></p>

8.3 Restore Default Account Policies

command	restoreaccountpolicytodefault
example URL	https://localhost/UMC/slwapi/restoreaccountpolicytodefault
HTTP Verb	POST
Parameters	none
Data	

Result

```
{  
  "version" : 0 ,  
  
  "operation" : "setaccountpolicyanswer",  
  
  "result" : 0 }
```

Important:

Status is 400 in case of error.

9 Function Rights

9.1 List Function Rights

command	rights
example URL	https://localhost/UMC/slwapi/rights
HTTP Verb	POST
Parameters	none
Data	
Result	<pre>{ "version" : 0 , "operation" : "rightsresult", "result" : 0, "rights" : [{ "name":"UM_ADMIN", "id" : 10001, "description" : "Administer UM Configuration" }, {"name":"UM_VIEW", "id" : 10002, "description" : "View UM Configuration" }, {"name":"UM_RESETPWD", "id" : 10003, "description" : "Reset user password" }, {"name":"UM_RA", "id" : 10004, "description" : "Login from Remote Authentication" }, {"name":"UM_UNLOCKUSR", "id" : 10005, "description" : "Unlock User" }, {"name":"UM_JOIN", "id" : 10006, "description" : "Create UM Server" }, {"name":"UM_ATTACH", "id" : 10007, "description" : "Create UM Agent" }, {"name":"UM_RINGMNG", "id" : 10008, "description" : "Join/Revoke an UM Server to the UM Ring" }, {"name":"UM_RESETJOIN", "id" : 10009, "description" : "Reset UM Domain Join" }, {"name":"UM_IMPORT", "id" : 10010, "description" : "Import UM Configuration" }, {"name":"UM_EXPORT", "id" : 10011, "description" : "Export UM Configuration" }, {"name":"UM_BACKUP", "id" : 10012, "description" : "Backup UM Configuration" }, {"name":"UM_EXPORTCK", "id" : 10013, "description" : "Export Claim Key" }, {"name":"UM_EXPORTDK", "id" : 10014, "description" : "Export Domain Key" }, {"name":"UM_ADSYNC", "id" : 10015, "description" : "Manage Active Directory Synchronization" }, {"name":"UM_VIEWELG", "id" : 10016, "description" : "View ELG Events" }]} <hr/>Important: Status is 400 in case of error.<hr/></pre>

10 Change Password

command	changepassword
example URL	https://localhost/UMC/slwapi/changepassword
HTTP Verb	POST
Content-Type	application/json
Parameters	
Data	<pre>{ "old_password" = "franz", "new_password" = "secret", }</pre>
	list of users to be created
Result	In case of success the object is returned including the assigned id.
	<pre>{ "Version": 0 //the version of the protocol "operation": "changepasswordanswer" // identify the operation called by the API "result": 0 //SL_RESULT }</pre>
	<hr/> Important: <u>Status is 400 in case of error</u> <hr/>

11 Authentication IISAPPPOOL identity - only for test

command	authlogged
example URL	https://localhost/UMC/slwapi/authlogged
HTTP Verb	POST
Content-Type	application/json
Parameters	
Data	
Result	<p>if the appool identity (umc_pool) is imported in UMC the authentication method return success. This method is only for internal use and only for test purpose.</p> <div style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;"><pre>{"version" : 0 , "operation" : "loginresult", "result" : 0}</pre></div> <hr/> <p>Important: Status is 400 in case of error.</p>

12 Set Identity Property

command	setidentityproperty
example URL	https://localhost/UMC/slwapi/setidentityprop
HTTP Verb	POST
Content-Type	application/json
Parameters	
Data	<pre>{ "propcode" = 22, "propname" = "", "propvalue" = "it-IT", }</pre>
This is an example to change the current logged user language property	
Result	<pre>{ "operation": "setidentitypropanswer" // identify the operation called by the API "result": 0 //SL_RESULT }</pre>
Important: Status is 400 in case of error	

propcode must be one of the following values:

SL_USER_COMMENT 0x05
SL_USER_LANGUAGE 0x16
SL_USER_DATA_LANGUAGE 0x17
SL_USER_EMAIL1 0x102
SL_USER_EMAIL2 0x103
SL_USER_EMAIL3 0x104

SL_USER_FIRST_NAME 0x105
SL_USER_INITIALS 0x106
SL_USER_LAST_NAME 0x107
SL_USER_PHONE 0x108
SL_USER_MOBILE 0x109

or, only for changing custom attributes:

SL_USER_CUSTOM_PROPERTY 0x2FF

The szPropName parameter must be used only when you want to set a custom attribute (it is the name of the custom property to set), in all other cases this parameter is ignored.