Forcepoint

User ID Service API

2.0 or higher

User Guide

Revision A

Contents

Introduction

This guide describes how to use the API for the Forcepoint User ID Service (User ID Service) and provides examples of its use.

The User ID Service collects data about groups, users, and associated IP addresses from Windows Active Directory (AD) Servers, Microsoft Domain Controllers, and Microsoft Exchange Servers.

When you integrate the User ID Service with another Forcepoint product, such as Forcepoint Next Generation Firewall (Forcepoint NGFW), you can use the user data from the User ID Service for access control and monitoring users.

The User ID Service database provides data about users, their group memberships, and mappings between users and IP addresses using a REST API on TLS-protected port 5000. You can use this API to add, modify, and delete data in the User ID Service database.

By default, mappings of users to IP addresses are valid for 6 hours. You can adjust the default expiration time using the Forcepoint User ID Service Configuration Utility (Configuration Utility).

For more information, see *How to integrate Forcepoint User ID Service with other Forcepoint products* in Knowledge Base article 14100.

Working with RESTful principles

The Forcepoint User ID Service API is a REST API that includes these features.

- The API is based on the HTTP protocol and is platform-independent.
- The User ID Service API supports JSON representations for each object.

Although the User ID Service API follows the architectural style of RESTful web APIs, this guide introduces only the basic concepts. For more information, see:

- **Representational state transfer** — https://en.wikipedia.org/wiki/Representational_state_transfer and the linked content

Requests

The Forcepoint User ID Service API supports several types of requests that affect objects in different ways.

You can perform these actions:

- List and read objects using GET requests.
- Create objects using POST requests.
- Modify objects using PUT requests.
- Delete objects using DELETE requests.

Read access to the User ID Service API on TLS-protected port 5000 is anonymous. Actions that only read objects do not require authentication. For actions that create, modify, or delete objects, the User ID Service API requires HTTP Basic Authentication. If the Authorization header is not present in POST /user, PUT /user, or DELETE /user requests, the request fails and returns an HTTP 401 UNAUTHORIZED response.

Status codes and error messages

When requests are made, they return HTTP response status codes.

For more information, the HTTP response status codes follow the principles outlined in https://en.wikipedia.org/wiki/List_of_HTTP_status_codes.

For an error message, the server also attempts to send relevant information in the response body.

Request payload and query parameters

REST operations contain specific content or support additional parameters.

- POST and PUT requests require a JSON payload in the request.
- GET and DELETE requests do not require a JSON payload in the request.
- Some GET requests, such as GET /users, support filtering arguments as parameters.

Requirements

Familiarize yourself with these requirements before you use the Forcepoint User ID Service API.

- Make sure that the domains that the User ID Service monitors have been configured using the Configuration Utility.
- Add a User ID Service API user using the Configuration Utility.
- Configure User ID Service API clients to trust the certificate or the certificate authority (CA) that signed the certificate for the User ID Service server
- Configure the local host firewall on the User ID Service server to allow incoming connections to TCP port 5000 from your API clients.

Add an API user for Forcepoint User ID Service

Use the Configuration Utility to add an API user.

Note

In Forcepoint User ID Service version 2.1 and higher, you must add an API user during the initial setup of the User ID Service. After the initial setup of the User ID Service, you can use the Configuration Utility to add additional API users.

In an HA configuration, the list of API users is synchronized between cluster members.

Steps

1) Log on to the server as root.
   You can also use `sudo` when you enter commands.

2) Enter the following command:

```
fuid-cfg api add-user
```

3) Enter a user name for the API user.

4) Enter a password for the API user.

Result

The API user is added to the configuration.

Trusted certificates for TLS communication

The Forcepoint User ID Service API uses a certificate for TLS communication on port 5000. API clients must trust the certificate or the CA that signed the certificate.

You must configure API clients to trust the CA that signed the certificate for the User ID Service server. If you use a self-signed certificate for the User ID Service server, you must configure the API clients to directly trust the self-signed certificate.

Working with the Forcepoint User ID Service API

You can use the Forcepoint User ID Service API to read, create, modify, and delete objects.

Note

In an HA configuration, the API client can connect to any available User ID Service cluster member.

Using GET /domains

GET /domains returns a list of domains as LDAP distinguished names.

Resource URL

Use the following format for requests:

```
https://<service address>/api/uid/v1.0/domains
```

Parameters

None. The request assumes that the directory service is Active Directory.

Return

A successful request returns a list of domains as LDAP distinguished names.

Example

The following request:

```
https://192.0.2.1:5000/api/uid/v1.0/domains
```

Returns the following response:

```
{
 "domains": [
 "dc=us, dc=company, dc=com",
 "dc=eu, dc=company, dc=com",
 "dc=latam, dc=company, dc=com",
 "dc=asiapac, dc=company, dc=com"
 ]

}
```

Using GET /users

GET /users returns a list of users for the domains that the Forcepoint User ID Service service is monitoring, and information about each user.

Resource URL

Use the following format for requests:

```
https://<service address>/api/uid/v1.0/users
```

Parameters

All parameters are optional.

Parameter	Description
domain	When you specify a domain, the request returns only users in the specified domain. If you do not specify a domain, the request returns a list of all users. Enter a fully qualified domain name (FQDN) in URL encoding (percent-encoding) format.
group	When you specify a group, the request returns only users that are members of the specified group. Enter the LDAP distinguished name (dn) for the group in URL encoding (percent-encoding) format.
ip_only	If set to "true", the request returns only users that have IP address information associated with them. If set to "false", the request returns a list of all users in the specified domains. The default value is "true".
networks	When you specify network ranges, the request returns only users that are associated with at least one IP address in the specified network ranges. If a user is associated with more than one IP address, the results show all IP addresses associated with the user. Enter a single network range in the format 1.2.3.4-1.2.3.55, or a comma separated list of network ranges. The list of network ranges must use URL encoding (percent-encoding) format.

Return

- A successful request returns a 200 HTTP status code and a list of user objects.
- A 400 HTTP status code might be returned if the network range is invalid, such as if the second IP address is lower than the first.

Example

The following request with a domain filter:

```
https://192.0.2.1/api/uid/v1.0/users?domain=us.company.com
```

Returns the following response:


```
{
 "users": [{
 "dn": "CN=John Garcia,OU=ou_devUS,OU=ou_all,DC=us,DC=company,DC=com",
 "changetype": "add",
 "sAMAccountName": "jgarcia",
 "NTLMIdentity": "US1\\jgracias",
 "mail": "jgarcia@us.company.com",
 "ipv4_addresses": [
 "192.0.2.12",
 "198.51.100.48",
 "203.0.113.141"
 ],
 "objectGUID": "9a151180-3cbe-4379-b109-494332bc5b03",
 "groups": [
 "CN=Domain Admins,CN=Users,DC=us,DC=company,DC=com",
 "CN=Domain Users,CN=Users,DC=us,DC=company,DC=com",
 "CN=IT Admins,CN=Users,DC=us,DC=company,DC=com"
 ],
 "timestamp": "1494892106.333844"
 }, {
 "dn": "CN=Bob Smith,OU=ou_devUS,OU=ou_all,DC=us,DC=company,DC=com",
 "changetype": "add",
 "sAMAccountName": "bsmith",
 "NTLMIdentity": "US1\\bsmith",
 "mail": "bsmith@us.company.com",
 "ipv4_addresses": ["203.0.113.132"],
 "objectGUID": "745414ad-813f-4c52-bb8a-e65f97397121",
 "groups": [
 "CN=Domain Users,CN=Users,DC=us,DC=company,DC=com",
 "CN=Guitarist,CN=Users,DC=us,DC=company,DC=com",
 "CN=Vocalist,CN=Users,DC=us,DC=company,DC=com"
 ],
 "timestamp": "1494892106.333844"
 }]
}
```


Using GET /user

GET /user returns a single user object.

Resource URLs

Use the following format for requests:

```
https://<service address>/api/uid/v1.0/user/<object guid>
https://<service address>/api/uid/v1.0/user/dn/<distinguished name>
https://<service address>/api/uid/v1.0/user/ntlm-identity/<DOMAIN\Username>
```

Parameters

At least one of the following parameters is required. All parameters must use URL encoding (percent-encoding) format.

Parameter	Description
dn	The LDAP distinguished name (dn) for the user.

Parameter	Description
ntlm-identity	The down-level logon name, in the format DOMAIN\UserName. For more information about user name formats, see https://docs.microsoft.com/en-us/windows/desktop/SecAuthN/user-name-formats#down_level_logon_name.
obj_guid	The LDAP object GUID for the user.

Return

A successful request returns a 200 HTTP status code and the user object that maps to the specified dn, ntlm-identity, or obj_guid.

Examples

There are multiple ways that you can retrieve a user.

The following request retrieves a user using the object GUID:

```
https://192.0.2.1/api/uid/v1.0/user/OCjIDqJml0aFwiTUc0O94w==
```

The following request retrieves a user using the distinguished name:

```
https://192.0.2.1/api/uid/v1.0/user/dn/CN%3DJohn%20Garcia%2COU%3Dou_devUS%2COU%3Dou_all%2CDC%3Dus
%2CDC%3Dcompany%2CDC%3Dcom
```

The following request retrieves a user using the down-level logon name (NTLM Identity):

```
https://192.0.2.1/api/uid/v1.0/user/ntlm-identity/US1%5Cjgarcia
```

All of the example requests return the following response:

```
{
 "dn": "CN=John Garcia,OU=ou_devUS,OU=ou_all,DC=us,DC=company,DC=com",
 "changetype": "add",
 "sAMAccountName": "jgarcia",
 "NTLMIdentity": "US1\\jgracias",
 "mail": "jgarcia@us.company.com",
 "ipv4_addresses": [
 "192.0.2.12",
 "198.51.100.48",
 "203.0.113.141"
 ],
 "objectGUID": "82764472-ac2e-47bc-a2cc-27c7c994be4b",
 "groups": [
 "CN=Domain Admins,CN=Users,DC=us,DC=company,DC=com",
 "CN=Domain Users,CN=Users,DC=us,DC=company,DC=com",
 "CN=IT Admins,CN=Users,DC=us,DC=company,DC=com"
 ],
 "timestamp": "1494892106.333844"
 }
```

Using POST /user

POST /user adds a user object to the Forcepoint User ID Service database.

Note

This request requires authentication.

Resource URLs

Use the following format for requests:

```
https://<service address>/api/uid/v1.0/user/<object guid>
https://<service address>/api/uid/v1.0/user/ntlm-identity/<DOMAIN\UserName>
```

Parameters

Either ntlm-identity or obj_guid is required. The URL specifies whether to add the user using the object GUID or the NTLM Identity. All parameters must use URL encoding (percent-encoding) format.

Parameter	Description
ntlm-identity	The down-level logon name, in the format DOMAIN\UserName For more information about user name formats, see https://docs.microsoft.com/en-us/windows/desktop/SecAuthN/user-name-formats#down_level_logon_name.
obj_guid	The LDAP object GUID for the user. If you do not specify an object GUID, one is automatically created and returned in the response. If you create a user object using a GUID, you must specify the dn or NTLMIdentity in the payload.

The payload of the request must be JSON data that contains the user attributes. For a list of user attributes, see the table of user object attributes.

The payload must specify either a dn or an NTLMIdentity.

Note

If there is already a user with the specified dn, object GUID, or NTLMIdentity, the request fails.

If you want to filter the user based on the domain, the payload must specify either a dn or a mail attribute. The domain to which the user belongs is determined based on the specified attribute:

- If the payload contains a dn attribute, the domain is determined based on the dn attribute.
- If the payload does not contain a dn attribute, the domain is determined based on the mail attribute.

Note

If you do not specify either of these attributes, the user does not appear in GET requests that return only users in the specified domains.

Return

- A successful request returns a 200 HTTP status code and the object GUID. If you specified an object GUID in the request, the same object GUID is returned. Otherwise, an automatically created object GUID is returned.
- A 409 HTTP status code is returned if there is already a user object with the specified dn, NTLMIdentity, or object GUID.
- A 400 HTTP status code is returned in the following cases:
  - In requests that use the object GUID, required parameters or attributes in the JSON payload are missing. You must specify the NTLMIdentity as a parameter in the URL or in the JSON payload.
  - In requests that use the down-level logon name (NTLMIdentity), the JSON payload is empty, or the ntlm-identity attribute in the JSON payload is different from the ntlm-identity parameter in the request URL.
- A 401 HTTP status code is returned if the Authorization header is missing or the credentials that you entered are incorrect.

Examples

The following request creates a user using the object GUID:

```
https://192.0.2.1/api/uid/v1.0/user/d8e3d58f-fbc9-4ba9-8206-925c3083ed7d
```

The following request creates a user using the down-level logon name (NTLM Identity):

```
https://192.0.2.1/api/uid/v1.0/user/ntlm-identity/US1%5Cjdoe
```

Example payload:

```
{
 "dn": "CN=Jane Doe,OU=ou_devUS,OU=ou_all,DC=us,DC=company,DC=com",
 "sAMAccountName": "jdoe",
 "NTLMIdentity": "US1\\jdoe",
 "mail": "jdoe@us.company.com",
 "ipv4_addresses": [
 "192.0.2.12",
 "198.51.100.48",
 "203.0.113.141"
 ],
 "objectGUID": "d8e3d58f-fbc9-4ba9-8206-925c3083ed7d",
 "groups": [
 "CN=Bass Players,CN=Users,DC=us,DC=company,DC=com",
 "CN=Domain Users,CN=Users,DC=us,DC=company,DC=com",
 "CN=Groovers,CN=Users,DC=us,DC=company,DC=com"
 ],
 }
```

The following request creates a user using the down-level logon name (NTLM Identity) and specifies a timeout value for the user entry when there is at least one IP address in the payload:

```
https://192.0.2.1/api/uid/v1.0/user/ntlm-identity/US1%5Cjdoe
```

Example payload:

```
{
 "dn": "CN=Jane Doe,OU=ou_devUS,OU=ou_all,DC=us,DC=company,DC=com",
 "sAMAccountName": "jdoe",
 "NTLMIdentity": "US1\\jdoe",
 "mail": "jdoe@us.company.com",
 "ipv4_addresses": [
 "192.0.2.12",
 "198.51.100.48",
 "203.0.113.141"
 ],
 "objectGUID": "d8e3d58f-fbc9-4ba9-8206-925c3083ed7d",
 "groups": [
 "CN=Bass Players,CN=Users,DC=us,DC=company,DC=com",
 "CN=Domain Users,CN=Users,DC=us,DC=company,DC=com",
 "CN=Groovers,CN=Users,DC=us,DC=company,DC=com"
 ],
 "timeout": "120"
}
```

All of the example requests return the following response:

```
{  "objectGUID": "d8e3d58f-fbc9-4ba9-8206-925c3083ed7d" }
```

Related reference
User object attributes on page 17

Using PUT /user

PUT /user adds, modifies, or deletes IP addresses and groups for an existing user entry.

Note

This request requires authentication.

Resource URLs

Use the following format for requests:

```
https://<service address>/api/uid/v1.0/user/<object GUID>
https://<service address>/api/uid/v1.0/user/ntlm-identity/<DOMAIN\Username>
```

Parameters

At least one of the following parameters is required. All parameters must use URL encoding (percent-encoding) format.

Parameter	Description
ntlm-identity	The down-level logon name, in the format DOMAIN\UserName. For more information about user name formats, see https://docs.microsoft.com/en-us/windows/desktop/SecAuthN/user-name-formats#down_level_logon_name.

Parameter	Description
obj_guid	The LDAP object GUID for the user.

The changetype parameter in the JSON payload specifies the action:

- add — Adds the specified IP addresses and groups from the JSON data into the user entry specified by the GUID. This action does not modify existing IP addresses or groups in the user entry.
- modify — Replaces the existing IP addresses or groups with the IP addresses and groups specified in the JSON data.
- delete — Removes the IP addresses or groups specified in the JSON data.

Note

If the specified IP addresses or groups do not exist, a request to delete them returns a 200 HTTP response code because the delete action expects that the specified attributes are not included in the user object.

Return

Note

This request fails if the specified object GUID or NTLMIdentity cannot be found, or if the IP address or group is not specified.

- A successful request returns a 200 HTTP status code and the object GUID.
- A 400 HTTP status code is returned if no IP address or group is specified in the JSON payload.
- A 404 HTTP status code is returned if the specified object GUID or NTLMIdentity cannot be found.
- A 401 HTTP status code is returned if the Authorization header is missing or the credentials that you entered are incorrect.

Example: Adding IP addresses and groups to a user

The following request adds IP addresses and groups to a user using the object GUID:

```
https://192.0.2.1/api/uid/v1.0/user/6cd757c5-e815-470c-beef-d17f8b967f8e
```

The following request adds IP addresses and groups to a user using the down-level logon name (NTLMIdentity):

```
https://192.0.2.1/api/uid/v1.0/user/ntlm-identity/US1%5Cjdoe
```

Example payload:

```
{
"objectGUID": "6cd757c5-e815-470c-beef-d17f8b967f8e",
"changetype": "add",
"ipv4_addresses": [
"192.0.2.1",
"192.0.2.2",
"192.0.2.3"
],
"ipv6_addresses": [
"2001:db8:a28b:14:8539:f8ab:493f:aba1",
"2001:db8:a28b:14:8539:f8ab:493f:aba2",
"2001:db8:a28b:14:8539:f8ab:493f:aba3"
],
"groups" : [
"CN=Test Users1,CN=Users,DC=example,DC=com",
"CN=Test Users2,CN=Users,DC=example,DC=com"
]
}
```

These example requests return the following response:

```
{  "objectGUID": "6cd757c5-e815-470c-beef-d17f8b967f8e" }
```

Example: Modifying IP addresses and groups for a user

The following request modifies IP addresses and groups for a user using the object GUID:

```
https://192.0.2.1/api/uid/v1.0/user/6cd757c5-e815-470c-beef-d17f8b967f8e
```

The following request modifies IP addresses and groups for a user using the down-level logon name (NTLMIdentity):

```
https://192.0.2.1/api/uid/v1.0/user/ntlm-identity/US1%5Cjdoe
```

Example payload:

```
{
"objectGUID": "6cd757c5-e815-470c-beef-d17f8b967f8e",
"changetype": "modify",
"ipv4_addresses": [
"192.0.2.4",
"192.0.2.5",
"192.0.2.6"
],
"ipv6_addresses": [
"2001:db8:a28b:14:8539:f8ab:493f:aba4",
"2001:dn8:a28b:14:8539:f8ab:493f:aba5",
"2001:dn8:a28b:14:8539:f8ab:493f:aba6"
],
"groups" : [
"CN=Test Users3,CN=Users,DC=example,DC=com",
"CN=Test Users4,CN=Users,DC=example,DC=com"
]
}
```

These example requests return the following response:

```
{  "objectGUID": "6cd757c5-e815-470c-beef-d17f8b967f8e" }
```

Example: Deleting IP addresses and groups from a user

The following request deletes IP addresses and groups from a user using the object GUID:

```
https://192.0.2.1/api/uid/v1.0/user/6cd757c5-e815-470c-beef-d17f8b967f8e
```

The following request deletes IP addresses and groups from a user using the down-level logon name (NTLMIdentity):

```
https://192.0.2.1/api/uid/v1.0/user/ntlm-identity/US1%5Cjdoe
```

Example payload:

```
{
"objectGUID": "6cd757c5-e815-470c-beef-d17f8b967f8e",
"changetype": "delete",
"ipv4_addresses": [
"192.0.2.1",
"192.0.2.3"
],
"ipv6_addresses": [
"2001:dn8:a28b:14:8539:f8ab:493f:aba1",
"2001:dn8:a28b:14:8539:f8ab:493f:aba3"
],
"groups" : [
"CN=Test Users1,CN=Users,DC=example,DC=com"
]
}
```

These example requests return the following response:

```
{  "objectGUID": "6cd757c5-e815-470c-beef-d17f8b967f8e" }
```

Using DELETE /user

DELETE /user deletes the map entries associated with the specified object GUID.

Note

This request requires authentication.

Resource URL

Use the following format for requests:

```
https://<service address>/api/uid/v1.0/user/<object guid>
```

Parameters

Parameter	Description
obj_guid (Required)	The LDAP object GUID for the user in URL encoding (percent-encoding) format.

The dn and NTLMIdentity attributes are not required. No JSON payload is needed.

Return

- A successful request returns a 200 HTTP status code and the same object GUID that was specified in the request.
- A 500 HTTP status code is returned if an error occurs when the server processes the data.
- A 404 HTTP status code is returned if the specified object GUID cannot be found or the user has already been deleted.
- A 401 HTTP status code is returned if the Authorization header is missing or the credentials that you entered are incorrect.

Example

The following request:

```
https://192.0.2.1/api/uid/v1.0/user/6cd757c5-e815-470c-beef-d17f8b967f8e
```

Returns the following response:

```
{  "objectGUID": "6cd757c5-e815-470c-beef-d17f8b967f8e" }
```

Using GET /groups

GET /groups returns all groups for the configured domain.

Note

This request can be resource-intensive. Because group information is not likely to change, we do not recommend using this request frequently.

Resource URL

Use the following format for requests:

```
https://<service address>/api/uid/v1.0/groups
```

Parameters

Parameter	Description
domain (Optional)	When you specify a domain, the request returns only groups in the specified domain. If you do not specify a domain, the request returns a list of all groups. Enter a the LDAP distinguished name (dn) of the group in URL encoding (percent-encoding) format.

Return

A list of group objects.

Example

The following request:

```
https://192.0.2.1/api/uid/v1.0/groups?domain=dc%3Dus%2Cdc%3Dcompany%2Cdc%3Dcom
```

Returns the following response:


```
{
 "groups": [
 {
 "NTLMIdentity": "\\Domain Admins",
 "changetype": "add",
 "dn": "CN=Domain Admins,CN=Users,DC=example,DC=local",
 "groups": "['CN=Administrators,CN=Builtin,DC=example,DC=local', 'CN=Denied RODC Password
 Replication Group,CN=Users,DC=example,DC=local']",
 "objectClass": "Group",
 "objectGUID": "9b7ee14d-6235-45e6-a563-bc293a6f2422",
 "sAMAccountName": "Domain Admins",
 "timestamp": "1494892106.309293"
 },
 {
 "NTLMIdentity": "\\DnsAdmins",
 "changetype": "add",
 "dn": "CN=DnsAdmins,CN=Users,DC=example,DC=local",
 "objectClass": "Group",
 "objectGUID": "341e3fdc-81ed-4ff1-8a38-c98c531f9665",
 "sAMAccountName": "DnsAdmins",
 "timestamp": "1494892106.342993"
 } ] }
```


Using GET /status

GET /status returns the last update, total number of domains, total number of users, and number of users in each domain as a status.

Resource URL

Use the following format for requests:

```
https://<service address>/api/uid/v1.0/status
```

Parameters

None.

Return

- A successful request returns a 200 HTTP status code and a list of status data.
- A 500 HTTP status code is returned if an error occurs on the server.

Example

The following request:

```
https://192.0.2.1/api/uid/v1.0/status
```

Returns the following response:

```
{
"status": {
"Last update": "Fri, 03 Nov 2017 16:22:08 GMT",
"Total domains count": 2,
"Total users count": 19,
"Users count per domain": {
"domain1.com": 1,
"domain2.com": 18
}
}
}
```

User object attributes

User objects include these attributes.

Attribute	Description	Example
dn	The LDAP distinguished name for the user.	`CN=John Smith,OU=ou_devUS,OU=ou_all,DC=us,DC=company,DC=com`
sAMAccountName	The logon name used to support earlier clients.	`jsmith`
NTLMIdentity	The down-level logon name, in the format DOMAIN\UserName. For more information about user name formats, see https://docs.microsoft.com/en-us/windows/desktop/SecAuthN/user-name-formats#down_level_logon_name.	`DOMAIN\jsmith`
mail	Email address.	`jsmith@us.company.com`

Attribute	Description	Example
ipv4_addresses (Optional)	A list of IPv4 addresses.	`"ipv4_addresses": [ "172.86.141.72", "203.64.92.48", "178.36.80.141" ]`
ipv6_addresses (Optional)	A list of IPv6 addresses.	`2001:0db8:85a3:0000:0000:8a2e:0370:7334`
objectGUID	The object GUID for the LDAP object is the unique identifier of a user.	`187712cf-81e2-49fe-93b3-a9a7836afe10`
groups	A list of group names.	`"groups" : [ "CN=Domain Users,CN=Users,DC=us,DC=company, DC=com", "CN=IT Admins,CN=Users,DC=us,DC=company, DC=com" ]`
changetype	The state of the object. The possible states are: ■ add — The changetype is add when the user is added from the AD or added using POST /user. ■ delete — The changetype is delete if the user is deleted from the AD or deleted using DELETE /user. ■ modify — The changetype is modify if the user record is modified in the AD or using PUT /user. ■ ip-add — The changetype is ip-add when the last modification was to add an IP address. The ip-add changetype is also returned from a GET /user request if IP addresses were added. ■ ip-delete — The changetype is ip-delete when the last modification was to delete an IP address. The ip-delete changetype is also returned from a GET /user request if IP addresses were deleted.	`"changetype": "add",`

Attribute	Description	Example
timestamp	The time when an update was last received for mappings between users and IP addresses, and group memberships. If there are no updates for mappings between users and IP addresses, and group memberships, the time stamp is the time that user information was polled from the directory service.	`1494892106.321318`

Group object attributes

Group objects include these attributes.

Field	Description	Example
dn	The LDAP distinguished name for the group.	`"CN=Domain Admins,CN=Users,DC=us,DC=company, DC=com"`
sAMAccountName	The logon name used to support earlier clients.	`Domain Admins`
NTLMIdentity	The down-level logon name, in the format DOMAIN\UserName. For more information about user name formats, see https://docs.microsoft.com/en-us/windows/desktop/SecAuthN/user-name-formats#down_level_logon_name.	`\\Domain Admins`
objectGUID	The object GUID for the LDAP object is the unique identifier of the group.	`2c14c3bb-bdfa-414b-bfd8-f5d83e69be1d`
groups	A list of groups that this group is a member of.	`"groups" : [ "CN=Domain Admins,CN=Users,DC=us,DC=company, DC=com", "CN=Domain Users,CN=Users,DC=us,DC=company, DC=com", "CN=IT Admins,CN=Users,DC=us,DC=company, DC=com" ]`

Field	Description	Example
changetype	The state of the object. The possible states are: ■ add — The changetype is add when the user record is added from the AD or if you create an entry using POST /user. ■ delete — The changetype is delete if the user record is deleted from the AD. ■ modify — The changetype is modify if an IP address is added or deleted for a user record, or if the user record is modified in the AD.	`"changetype": "add",`
timestamp	The time when an update was last received.	`1494892106.321318`

Published 28 October 2020