User Authentication for APIs

3 months ago Updated

Providing users with role access and assigning a facility to a user provides greater control and security. When a user is assigned to a facility, the user can issue API calls for the assigned facility based on his role. A role grants the user the ability to perform certain tasks. You can manage roles and assign facility to a user in the *User Management* section in the Manage *Administration* menu. Refer to the list of following articles:

- User Role Permissions
- Add Users and User Roles in Manage
- Assign a Facility to a User

When an API request is made, if the user role provides permission to the requested API and the user has access to the assigned facility, the request will be authorized and allowed to be complete. For example, users can request data from the sensors on the floor or area in the facility to which they have access. The API call returns a permission error if the user does not have access to the facility or the user role does not provide the permission to view data. However, note that the user role authorization feature is not available in previous releases. This feature has been added from version 3.9.13 onwards. Refer to the User Role Permissions for APIs.

Users must be authenticated to send or receive API requests to and from Manage. For authentication, send the following headers along with the REST API.

- *API key* -- Unique identifier for the user (this is the user name, for example, Bob and the generated API key copied from the EM system).
- *Timestamp* -- Time, date, and day of the API call. This is included to avoid replay attacks.
- *Authorization* -- SHA-1 authorization key (Calculated using the API key and timestamp).

Generating the API Key

To generate an API key for a user, see Generate API Key. Then, determine the timestamp (ts), authorization token as explained below, and send the headers along with the REST API call.

For example, user Bob is assigned the following values:

- Username: bob
- API Key: 6eb6f07fd09b18dd61dd353dfb669820e7859cd3 (The API Key copied from Manage)

Time Stamp and SHA-1 Authorization

Calculate timestamp and SHA-1 authorization values for the user (for example, bob) as follows:

1. Use the formula below to calculate timestamp (ts):

   ts=echo $(($(date +%s%N)/1000000))
   For example, if today's GMT date and time was Thursday, March 3, 2016, 7:36:51.032 PM, the timestamp would be *1457033811032*

- ts: *1457033811032*

2. Use the following command to calculate SHA-1 authorization in Linux, for example.

   SHA1="$(echo -n "$username$apikey$ts" | sha1sum -t | awk '{print $1}')".

- Authorization: *e20ac2c963ccfacf23a1f70287286443820e66d1*

For API authentication, send the following headers along with the REST API call:

APIkey: *bob* (Note: The APIkey here is the username)

Authorization: *e20ac2c963ccfacf23a1f70287286443820e66d1*

ts: *1457033811032*

API Example:

```
:~$ curl -s --get -H "ApiKey:  bob" -H "Authorization:
e20ac2c963ccfacf23a1f70287286443820e66d1" -H "ts:1457033811032" -H "Accept:
application/xml" -H "Content-Type: application/xml" -k
https://em_ip_address/ems/api/org/em/v1/energy -v -k
 * Hostname was NOT found in DNS cache
 *   Trying 127.0.0.1...
 * Connected to localhost (127.0.0.1) port 443 (#0)
 * successfully set certificate verify locations:
 *   CAfile: none
   CApath: /etc/ssl/certs
 * SSLv3, TLS handshake, Client hello (1):
 * SSLv3, TLS alert, Client hello (1):
 * SSLv3, TLS handshake, Server hello (2):
 * SSLv3, TLS handshake, CERT (11):
 * SSLv3, TLS handshake, Server key exchange (12):
 * SSLv3, TLS handshake, Server finished (14):
 * SSLv3, TLS handshake, Client key exchange (16):
 * SSLv3, TLS change cipher, Client hello (1):
 * SSLv3, TLS handshake, Finished (20):
 * SSLv3, TLS change cipher, Client hello (1):
 * SSLv3, TLS handshake, Finished (20):
 * SSL connection using ECDHE-RSA-AES256-GCM-SHA384
 * Server certificate:
 * subject: C=US; ST=California; L=Sunnyvale; O=Enlighted Inc.; OU=Manage
 * start date: 2015-12-15 06:37:22 GMT
```

```
* expire date: 2040-12-08 06:37:22 GMT
* issuer: C=US; ST=California; L=Sunnyvale; O=Enlighted Inc.; OU=Manage
* SSL certificate verify result: self signed certificate (18), continuing anyway.
> GET /ems/api/org/facility/v1/energy/1 HTTP/1.1
> User-Agent: curl/7.35.0
> Host: localhost
> ApiKey:  bob
> Authorization: e20ac2c963ccfacf23a1f70287286443820e66d1
> ts:  1457033811032
> Accept: application/xml
> Content-Type: application/xml
>
< HTTP/1.1 200 OK
< Date: Thu, 03 Mar 2016 19:42:55 GMT
* Server Apache-Coyote/1.1 is not blacklisted
< Server: Apache-Coyote/1.1
< Content-Type: application/xml
< Content-Length: 234
< Set-Cookie: JSESSIONID=8CE1F42FCBCCEE5AB69175D45673951F; Path=/ems/; HttpOnly
< Via: 1.1 127.0.0.1
< Vary: Accept-Encoding
```