Authentication

View the `/auth` section information in the WebAPI documentation for full details on the authentication methods of the eGauge meter WebAPI.

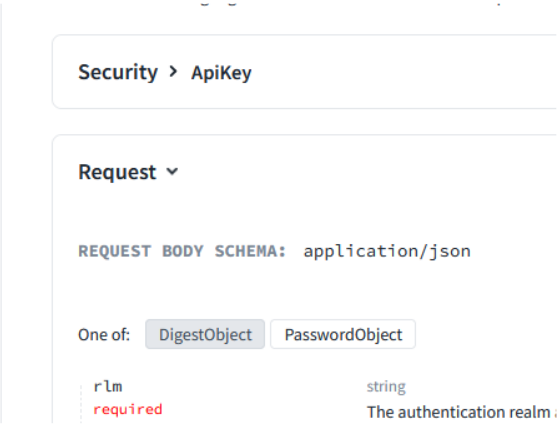
The eGauge WebAPI uses JSON web token (JWT) authentication for all interactions. An "Authorization" header must be provided with WebAPI requests in the format of `Authorization: Bearer JWT` where `JWT` is a valid JSON web token.

Tokens typically expire after 10 minutes and need to be renewed periodically.

There are two methods to obtaining a token:

1. Digest Object. This is the recommended method for obtaining a JWT and is described in this document. This method is similar to the HTTP Digest authentication method.
2. Password Object. This method sends the credentials as plaintext, and therefore requires a secure connection, e.g., HTTPS to the local IP address of the meter. This method will return an error if attempted over the eGauge proxy server, even if HTTPS is used.

You may find more information about the authentication methods by clicking the object you wish to use in the WebAPI documentation `/auth` section:



Digest Authentication workflow

1. Send a GET to /auth/unauthorized to get a 401 to get the realm (`rlm`) and server nonce (`nnc`)
2. Generate a client nonce (`cnnc`)
3. Calculate `hash` in the format:
   `ha1 = MD5(usr:rlm:pwd)`
   `hash = MD5(ha1:nnc:cnnc)`
   where `usr` and `pwd` are a valid user and password on the meter

4. Send `rlm`, `usr`, `nnc`, `cnnc` and `hash` to /auth/login for the token

Digest Authentication in Python

eGauge Systems provides a Python library with helper functions to deal with authentication and other interactions. See the WebAPI introduction page for more information.

```
#!/usr/bin/env python3

# Example Python script obtaining a JSON web token (JWT) from a meter's WebAPI.
# JWTs are needed for any interactions with the meter's JSON-based WebAPI.

# eGauge provides a Python library that handles authentication automatically and
# provides additional helper functions. It may be found on Bitbucket or PyPi
# https://bitbucket.org/egauge/python/src/master/egauge/
# https://pypi.org/project/egauge-python/

# Main WebAPI documentation: https://egauge.net/support/webapi

import requests
import hashlib
from secrets import token_hex

# meter and credential information
URI = "https://eGauge67385.d.egauge.net"
USER = "admin"
PASS = "as$kS2345da2@4vK9"



# get realm (rlm) and server nonce (nnc):
auth_req = requests.get(f"{URI}/api/auth/unauthorized").json()
realm = auth_req["rlm"]
nnc = auth_req["nnc"]

cnnc = str(token_hex(64)) # generate a client nonce (cnnc)

# generate our hash
# ha1 = MD5(usr:rlm:pwd)
# hash = MD5(ha1:nnc:cnnc)
ha1_content = f"{USER}:{realm}:{PASS}"
```

```
ha1 = hashlib.md5(ha1_content.encode("utf-8")).hexdigest()

hash_content = f"{ha1}:{nnc}:{cnnc}"
hash = hashlib.md5(hash_content.encode("utf-8")).hexdigest()

# Generate our payload
payload = {
    "rlm": realm,
    "usr": USER,
    "nnc": nnc,
    "cnnc": cnnc,
    "hash": hash
}

# POST to /auth/login to get a JWT
auth_login = requests.post(f"{URI}/api/auth/login", json=payload).json()

rights = auth_login["rights"] # rights this token has (save, control, etc)
jwt = auth_login["jwt"] # the actual bearer token

print(f"Got token with rights {rights}.")

# We can verify this token works.
# Add an authorization header with our token and make a request
headers = {"Authorization": f"Bearer {jwt}"}

api_request = requests.get(
    f"{URI}/api/config/net/hostname",
    headers=headers,
)

# {'result': 'eGauge67385'}
print(api_request.json())

# This token may be used until it expires, in which case a 401 response will be
# returned, to which this process can be reperformed.
```

Digest Authentication with Bash

This bash script uses curl and jq to obtain a JWT for use with the WebAPI.

```
URI="https://eGauge67385.d.egauge.net"
USER="admin"
PASS="as$kS2345da2@4vK9"

auth_req=$(curl -s "$URI/api/auth/unauthorized")
rlm=$(jq -r '.rlm' <<< $auth_req)
nnc=$(jq -r '.nnc' <<< $auth_req)
cnnc=$(openssl rand -hex 64)

ha1=$(echo -n "$USER:$rlm:$PASS" | md5sum | cut -f1 -d" ")
hash=$(echo -n "$ha1:$nnc:$cnnc" | md5sum | cut -f1 -d" ")

auth_login=$(curl -s -X POST "$URI/api/auth/login" \
    -H "Content-Type: application/json" \
    -d "{\"rlm\": \"$rlm\", \"usr\": \"$USER\", \"nnc\": \"$nnc\", \"cnnc\": \"$cnnc\", \"hash\": \"$hash\"}")

jwt=$(jq -r '.jwt' <<< $auth_login)

api_request=$(curl -s "$URI/api/config/net/hostname" -H "Authorization: Bearer $jwt")

echo $api_request
```

Please visit kb.egauge.net for the most up-to-date documentation.