Introduction

> The WebAPI documentation may be found at egauge.net/support/webapi.

> The Web API was introduced in firmware v.4.2, however eGauge Systems periodically releases new firmware versions with bug fixes, enhancements, and new features to the WebAPI. It is recommended to use the latest eGauge firmware version. **Click here** for information about checking and upgrading eGauge meter firmware.

> Be sure to select the appropriate firmware version in the WebAPI documentation for your meter in the upper left-hand corner of the documentation system. By default, the WebAPI documentation defaults to information about the latest stable firmware release.

The eGauge WebAPI is a JSON-based API framework that allows for automated meter configuration, recorded data retrieval, status information, and more.
eGauge Systems provides a Python library to assist with authentication and other interactions with the eGauge meter WebAPI and can be installed from from PyPi (pip) or Bitbucket (source code) .

Getting Started With Python

To make it easy to get started, eGauge provides an open source Python package. It can be installed with the command:

```
pip install egauge-python
```

With this package installed, accessing an eGauge meter becomes very simple. For example, to fetch the hostname of the meter, you could use:

```
from egauge import webapi

URI = "https://DEV.egaug.es" # replace DEV with meter name
USR = "USER" # replace USER with user name
PWD = "PASS" # replace PASS with password

dev = webapi.device.Device(URI, webapi.JWTAuth(USR,PWD))

print("hostname is " + dev.get("/config/net/hostname")["result"])
```

The package also contains various convenience classes to read meter data, capture waveform samples, convert between physical units, and so on.

The official GIT repository for this package is at https://bitbucket.org/egauge/python/. Various code examples can be found in the examples directory.

Getting started without the eGauge Python library

Check out the support library page on WebAPI Authentication for examples on authentication.

Common WebAPI service descriptions

See the WebAPI documentation for full details and all service endpoints available. The endpoints below are only several commonly accessed endpoints that retrieve data and configuration.

/auth

Service used to obtain or invalidate a JSON web token used for authenticating with the WebAPI.

/config

The `/config` service allows you to read and write configuration to the meter.

/register

The `/register` service provides instantanoues and historical *register* data that is recorded on the meter.

/local

The `/local` service provides instantaneous information derived from configured channel inputs, including RMS value, mean value, frequency value and instantanoues power and energy values if power registers are configured.

For historical or newly generated register data, use the `/register` service.

Other WebAPI service descriptions

Below are the additional service endpoints not listed above. See the WebAPI documentation for full details and all service endpoints available.

/capture

The `/capture` service is used for obtaining raw waveform data from the inputs. To obtain normal RMS, mean or frequency from the sensors directly, use the `/local` service. For obtaining stored or newly generated *register* data, use the `/register` service.

/cmd

Service used to send commands to the meter such as reboot or firmware upgrade.

/ctid

Service used to read or configure CTid sensors from a meter or flash the LED on the sensor (EG4xxx only).

/ctrl

Service used for controlling supported remote devices, such as the PRM3 relay module.

/log

Service used to access syslog or kernel logs. Syslog must be enabled through the `/config` service endpoint first.

/lua

Service used for managing Lua scripts on the meter.

/providers

Service used for obtaining information about third-party providers that support meter services such as push data sharing, push alerts, and tariffs.

/store

Service used for storing and retrieving preferences and other settings typically used in the user's web browser interface.

/remote

Service used for configuring remote devices on the meter.

/sys

Service used to obtain system information such as firmware version, uptime, serial number, meter model, and more.
