Elkor Meter Documentation

The user manual is very useful in explaining how to use the power meter and the web server.
User Manual: https://www.elkor.net/pdfs/WattsOn-Mark_II_Manual_Complete.pdf
The web server is explained in section 8 (starting on page 67).

Accessing the Web Server

To access the web server you have to open a browser and type in the power meter's IP address. Figure 1 shows an example of the IP address. Figure 2 shows what the front page looks like when it first loads.
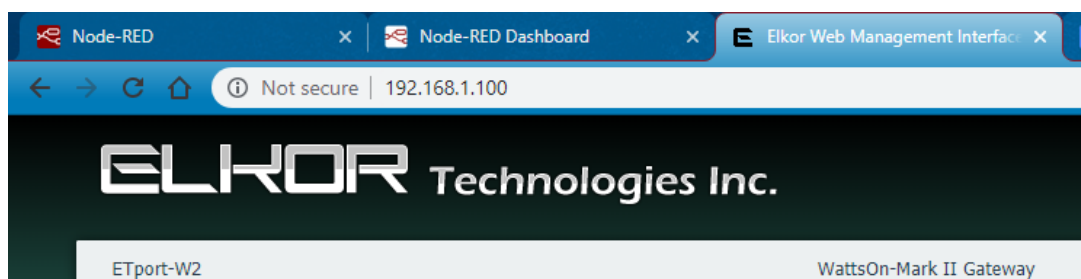

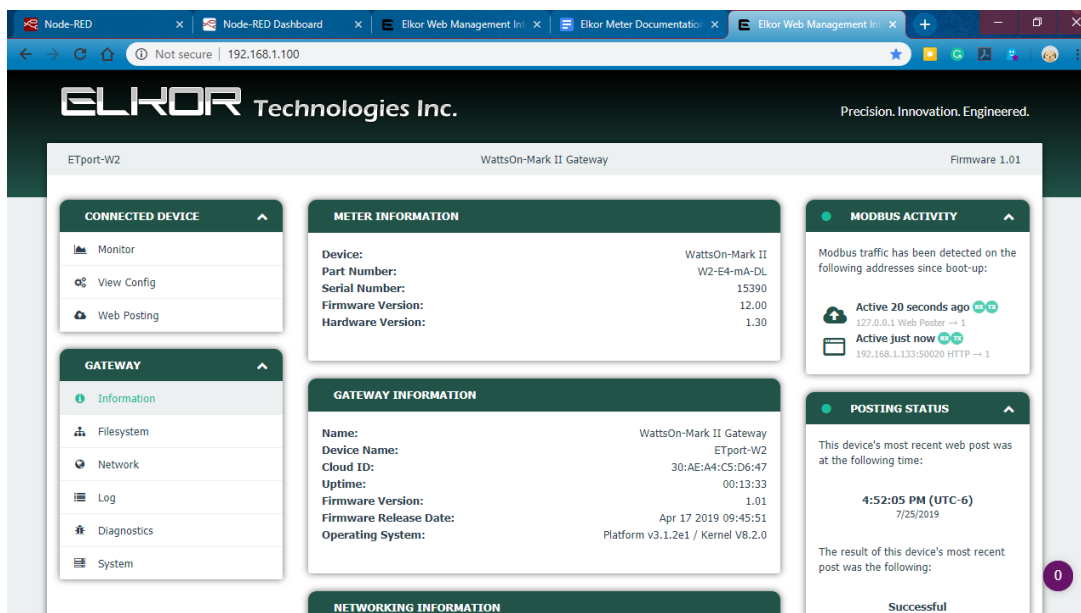**Figure 1:** Accessing the Web Server for a Meter

Figure 2: Front Page of Web Server

The front page gives basic identification information about the device and some information about the network it is on if you scroll down a little bit.

On the left side of the front page, there are a few different pages which is where more information as well as ways to configure the posting and the file output. Most of the tabs are self explanatory for what they contain. On the right side, the posting status and the devices activity inform you of the device's activity.

NOTE: When the page asks for a username and password to access the other tabs, the default values are the following:

Username: **admin**

Password: **admin**

The password can be changed via the **System** tab.

Editing the Posting Location and Editing the Output File

To configure posting, http authentication, and which output template to use, go to tab on the left side labeled **Web Posting.** In the middle there should be a panel as shown below in Figure 3.

Figure 3: Web Posting Configuration

The posting location can be configured with the first field, in have it currently posting to Node-Red via its IP address in the picture. When editing the posting location, you can also set the time between posts, with a minimum time of 1 minute per post. The http username and password can be set and changed in the next field. The third field allows you to choose which

output template should be used, it is initially set to a default JSON file, which probably already has all the information that you would need. If a format other than JSON is desired, you just need to upload file and choose it here.

To upload or delete files, click the **Filesystem** tab on the left side. The panels in the middle will tell you how much space is used on the meter as well as the list of files uploaded onto the meter. Here you can edit, delete, or upload files as necessary.

Editing File Post Headers and Other Configuration Settings

If there are headers that need to be added other than http authentication, there is a way to do so in the Elkor meter's advanced configuration. Go to the **System** tab on the left, then go to **Advanced Configuration**.This gives a list of various settings on their current and default values (if they have any). Figure 4 shows the advanced configuration page.
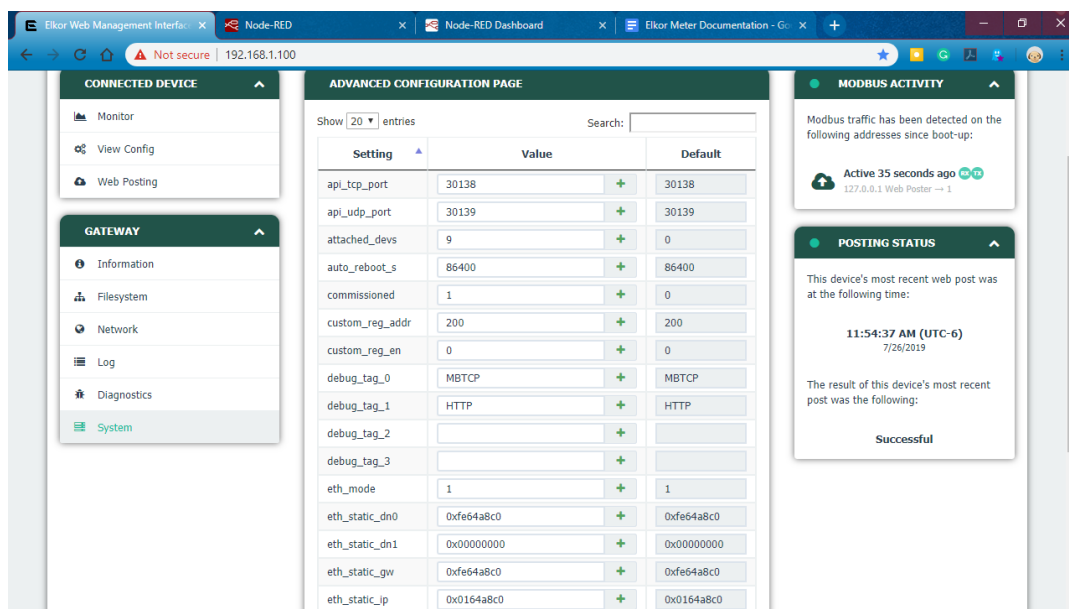

Figure 4: Elkor Meter Advanced Configuration Page

If you use the search bar, you can search *header* which gives two results, *http_headers* and *post_headers*. The advanced configuration page gives a large number of settings for the meter which can be configured as necessary.

Sending files to S3

Currently, our approach is to upload directly from the meter to and S3 bucket. While this is the simplest approach, it has a couple of drawbacks:
- We can't specify security credentials on the upload.
- We would have to create a CORS config with an Allowed List of ip addresses for all the meters to secure, which will change with each building.
- We cannot post-process the data at all

Where to upload:
- **PUT** request
- **https://data-elkor-test.s3.amazonaws.com/<NAME_OF_FILE>**
- *Replace the <NAME_OF_FILE> with something dynamic like a timestamp*
- **Content-Type: application/x-www-form-urlencoded** if needed

Version 2.0

The final version of this should be a small rails app that acts as a proxy. We could set up multiple endpoints that accept a file, then could use AWS security credentials to upload the file to the bucket. The app would have multiple endpoints for each type of meter and could normalize the naming of the files before uploading.

Initial Github repo — https://github.com/studio-otto/slp-meter-proxy

Potential challenges:
- Setting up a queue for when a file fails
- Testing loads on app

Heroku Microservice

API URL for Elkor meter to post to:

https://slp-meter-proxy.herokuapp.com/elkor