**Creating an information web page that can be picked up using the rig’s URL.**

**TODO** - implement this as a proper oref0 script that can be installed by oref0-setup

This allows you to extract data from the various files that OpenAPS creates and access the locally from the phone that is connected to the rig, giving a full information set.

Is this only for android, or does it work for iphone too?

Firstly, you need to set up the script that will do this for you. An example is shown below:

Where and how to do this? Run each line separately, or all together? From root, or from myopenaps? For example, I didn not have an enact/index.html to remove (maybe a note about which ones everyone has versus does not.. if it is known)

rm **~/**myopenaps**/**enact**/**index**.**html touch **~/**myopenaps**/**enact**/**index**.**html (cat **~/**myopenaps**/**enact**/**smb**-**enacted**.**json **|** jq **-**r **.**timestamp **|** awk '{print substr($0,12,5)}') **>>** **~/**myopenaps**/**enact**/**index**.**html (cat **~/**myopenaps**/**enact**/**smb**-**enacted**.**json **|** jq **-**r **.**reason) **>>** **~/**myopenaps**/**enact**/**index**.**html (echo **-**n 'TBR: ' **&&** cat **~/**myopenaps**/**enact**/**smb**-**enacted**.**json **|** jq **.**rate) **>>** **~/**myopenaps**/**enact**/**index**.**html (echo **-**n 'IOB: ' **&&** cat **~/**myopenaps**/**enact**/**smb**-**enacted**.**json **|** jq **.**IOB) **>>** **~/**myopenaps**/**enact**/**index**.**html (echo **-**n 'Edison Battery: ' **&&** cat **~/**myopenaps**/**monitor**/**edison**-**battery**.**json **|** jq **-**r **.**battery **|** tr '\n' ' ' **&&** echo '%') **>>** **~/**myopenaps**/**enact**/**index**.**html (echo **-**n 'Insulin Remaining: ' **&&** cat **~/**myopenaps**/**monitor**/**reservoir**.**json) **>>** **~/**myopenaps**/**enact**/**index**.**html

You may need to adjust the values in '{print substr($0,12,5)}' - whilst I know these work on the rigs I have set them up on, other’s have had better results with {print substr($0,13,5)}'

What do the variables here do? Hard to know how to adjust without know what print substr means.

It can be set up where you choose, either in your openaps directory or at root. Maybe this belongs at the beginning?

You will also need to start up the SimpleHTTPserver service that is already installed on jubilinux in the location you will place your file. This is done by adding the following line to your Cron:

Which cron? I have multiple. Maybe include the command to get to cron, and if it matters where you add this. I added it in the end, but not sure if it was supposed to be somewhere else.

Also, are numbers in blue specific to rig, or you can cut or paste? Including an instruction of whether to cut and paste, or fix information for your rig is helpful.

@reboot cd **/**root**/**myopenaps**/**enact **&&** python **-**m SimpleHTTPServer 1337

The final thing to do is to make sure the script runs regularly to collect the data and publish it. This requires an additional cron line:

So I can cut and paste just like this?

@reboot cd **/**root**/**myopenaps**/**enact **&&** python **-**m SimpleHTTPServer 1337 **\*/**5 **\*** **\*** **\*** **\*** (bash **/**root**/**http**.**sh) 2**>&**1 **|** tee **-**a **/**var**/**log**/**openaps**/**http**.**log

**\*/**5 **\*** **\*** **\*** **\*** (bash **/**root**/**http**.**sh) 2**>&**1 **|** tee **-**a **/**var**/**log**/**openaps**/**http**.**log

In this case the script is running from the /root directory and I am publishing to the ~/myopenaps/enact directory.

To access this from an iphone browser, enter something like the following: <http://172.20.10.x:1337/index.html>. Make sure to change the x to your rig’s ip address. and you should receive an unformatted html page with the data in it. If you want to improve the output for a browser, the script can be modified to generate html tags that will allow formatting and could provide colouring if various predicted numbers were looking too low.

Does this only work if rig and phone are on the same network? I know this seems obvious, but for the less technically inclined, it would be a helpful instruction

On Android, you can download http-widget (https://play.google.com/store/apps/details?id=net.rosoftlab.httpwidget1&hl=en\_GB) and add a widget to your home screen that will display this data.

If you use a Samsung Gear S3 watch, you can use the above http-widget with Wearable Widgets (http://wearablewidgets.com) to view what OpenAPS is doing locally, without internet connection.