Stats and Logging

Stats and Logging

By default dokuwiki tracks changes and stores metadata inside /data/meta and /date/media-meta. This is accessible through various dokuwiki plugins. For example the changes plugin....

Recent Changes

- 101 - Arduino [Step 2]
- Art of Projection
- Introduction to Neopixels
- 101 - Intro to 3D Modelling [Workshop presentation]
- Recording Studio Induction [Safety]
- 3dprinting101 ↻ Page name changed from workshops:public:3dprinting101 to workshops:public:3dmodelling101
- Recording Studio [Table]
- Laser cut covers, hand sewn spines - Participant resource page
- Laser Cut, Hand-Bound Books: Crafting the Grumpus' Treasured Tomes
- 2022 Bookbinding - Laser cut covers, hand sewn spines

Using Access Logs

If we want more detail, we can use the logstats plugin to generate a server log for our dokukwiki.

Then we can use goaccess to generate some pretty pictures, as well as export boring CSVs/JSON. As a command line tool, goaccess will also take input from pipes, which lets us use POSIX utilities to get what we want.

Installing Goaccess

Follow the instructions on the go access site to install from a repo

```
$ echo "deb http://deb.goaccess.io/ $(lsb_release -cs) main" | sudo tee -a
/etc/apt/sources.list.d/goaccess.list
$ wget -O - https://deb.goaccess.io/gnugpg.key | sudo apt-key add -
$ sudo apt-get update
$ sudo apt-get install goaccess
```

Config File and Browser List

The config file default installs to /etc/goaccess/goaccess.conf but for some reason, goaccess expects

it in /etc/goaccess.conf. Copy it over and to the same for the browser.list file while you are at it. Set the time and date formats to Apache/NGINX and the log type to COMBINED and we should be good to start.

Parsing the access.log

In this case I've copied the access log from /data/meta/access.log to work on it. We want:

- quarterly log
- bots and crawlers removed
- no internal IP address

so we are going to use a combination of tools for this. First up, lets use sed to grab the date range we want

```
sed -n '/1\/Jul\/2019/,/30\/Sep\/2019/ p' access.log
```

then grep with the -v option to exlude bots and dynomapper (this could be a single grep)

```
grep -i -v  --line-buffered 'bot' | grep -i -v --line-buffered 'dyno'
```

Finally let run goaccess, excluding our local IP range, ignoring crawlers and output to a html file.

```
goaccess -e 192.168.0.0-192.168.254.254 --ignore-crawlers -a -o
q1report.html
```

Running all these commands piped (for the next quarter) we get:

```
sed -n '/1\/Oct\/2019/,/31\/Dec\/2019/ p' access.log | grep -i -v  --line-
buffered 'bot'| grep -i -v --line-buffered 'dyno' | goaccess -e
192.168.0.0-192.168.254.254 --ignore-crawlers -a -o q2report.html
```

To see these logs just copy the resulting html to your web server root. You will end up with something like:
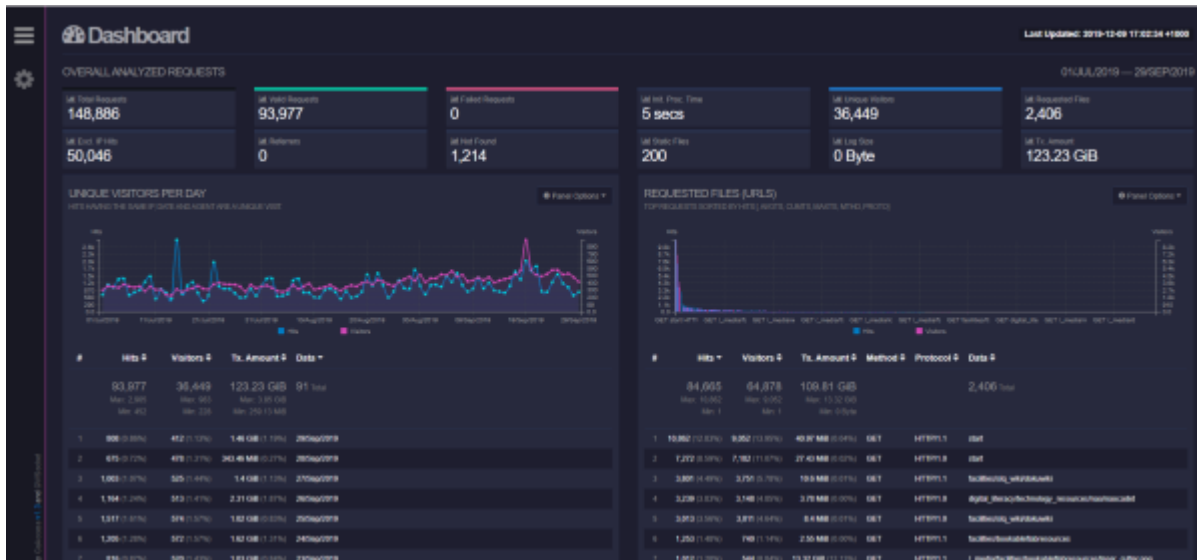
This is pretty, but a more useful output would be csv. A csv output can be expand to include any number of record (in the config) so we can use it to get a sense of the static files downloaded, which can also be set to include only the types of files we are interested in (also in the config). To get csv output, just change the filetype of the output i.e q2report.csv

Accessing Metadata

To work out how many pages have been created, we need to go back to dokuwiki's metadata. What we want is the metadata stored in .changes file for each page and media file, that was created in our date range, and not created by one of our team. The changes file on an newly created page inside /data/meta/ looks like:

```
1487560052 192.168.6.90    C    workshops    user    created        18942
```

"1487560052" is the timestamp in [unix time](#) the second is the IP address, "C" means created, and "user" is our users name. Thats all we need.

We only need the first line of each file called .changes. We can do this with the head command.

```
head  -1 ./*changes
```

Next we want to narrow our selection to files created in our date range. A quick check of https://www.epochconverter.com/ will give us the date range we want, which is 1569852000 - 157780079. We can use awk to match this.

```
awk '($1+0)>1569852000 && ($1+0)<1577800799'
```

Then we want to filter out our internal users with grep, and use the -c option to tally the output.

```
grep -v -c --line-buffered user
```

Finally lets turn on the globstar in our shell so we can use head recursively.

```
shopt -s globstar
```

Now our piped commands look like this:

```
head -1 **/*.changes |  awk '($1+0)>1569852000 && ($1+0)<1577800799'| grep -
v -c --line-buffered mick
```

This gives use the pages created in the date range specified. Do find the media created, we can run the same command in the /data/media-meta directory, grepping for our media types.