Creation of Access Logs

We develop the website in node js ad mongo db environment. Before understanding access log creation we need to understand what is access log why it is used in our case.

What is Access Logs ?

An access log is a list of all the requests for individual files that people have requested from a Web site. These files will include the HTML files and their imbedded graphic images and any other associated files that get transmitted. The access log (sometimes referred to as the "raw data") can be analyzed and summarized by another program.

In general, an access log can be analysed to tell you:

- The number of visitors (unique first-time requests) to a home page
- The origin of the visitors in terms of their associated server's domain name (for example, visitors from .edu, .com, and .gov sites and from the online services)
- How many requests for each page at the site, which can be presented with the pages with most requests listed first
- Usage patterns in terms of time of day, day of week, and seasonally

Access log keepers and analyzers can be found as shareware on the Web or may come with a Web server.

For we can forward logs to DNIF environment by 2 ways

1. By realtime streaming
2. By posting logs in json format using **/json/receive** service

***Note** : For real time streaming of your access log in dNIF environment we have to Create that log in **apache** format only. Because DNIF support apache log parsing only. Otherwise we have to request company to write parser for us for which company take charge from client company.*

- **Creation of access log in apache format in Nodejs environment**

Format for apache access log is given below. This access log format also called as CLF access log format

127.0.0.1 - - [05/Feb/2012:17:11:55 +0000] "GET / HTTP/1.1" 200 140 "-" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/535.19 (KHTML, like Gecko) Chrome/ 18.0.1025.5 Safari/535.19"

LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-agent}i\"" combined

- %h is the remote host (ie the client IP)
- %l is the identity of the user determined by identd (not usually used since not reliable)
- %u is the user name determined by HTTP authentication
- %t is the time the request was received.
- %r is the request line from the client. ("GET / HTTP/1.0")
- %>s is the status code sent from the server to the client (200, 404 etc.)
- %b is the size of the response to the client (in bytes)
- Referer is the Referer header of the HTTP request (containing the URL of the page from which this request was initiated) if any is present, and "-" otherwise.
- User-agent is the browser identification string.

The complete(?) list of formatters can be found here. The same section of the documentation also lists other common log formats; readers whose logs don't look quite like this one may find the pattern their Apache configuration is using listed there.

In node JS environment we have to do following things

A. We have to install the **"accesslog.logger"** npm package. From https://www.npmjs.com/package/apache-like-accesslog

B. We have to add code in **app.js** given below.

```
var accesslog = require('apache-like-accesslog');

app.use(accesslog.logger);
```

```
accesslog.configure({
        format: 'JSON',
        directory: 'logs',
        filename: ‘access.log’});
```

C. Now access logs are generated in **/logs/access.log** file in apache format. The access logs generated in given below.

```
::ffff:192.168.0.28 - - [24/May/2018:15:33:13] 8260 "POST /login
HTTP/1.1" 200 215
::ffff:192.168.0.28 - - [24/May/2018:15:33:14] 2764 "GET /
getOverallAnalytics HTTP/1.1" 200 384
```

D. You can store this access.log file in any where in your instance.

- **Creation of access log in JSON format in Nodejs environment**

For posting the logs in DNIF platform using inbuilt **/json/receive/** service. We need to post that data in JSON format. Before that we need to understand the data in JSON format.

What is JSON formatted data.

JSON (JavaScript Object Notation) is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate. It is based on a subset of the JavaScript Programming Language, Standard ECMA-262 3rd Edition - December 1999. JSON is a text format that is completely language independent but uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, Python, and many others. These properties make JSON an ideal data-interchange language.

Please refer given like for more information

https://www.json.org/

In node js environment to create access as well as error log in in JSON format we need to do following things

A. We have to install the **"express-winston"** npm package. From https://www.npmjs.com/package/express-winston

B. We have to add code in **app.js** given below.

```
var winston = require('winston'),
expressWinston = require('express-winston');


expressWinston.requestWhitelist.push('body');

app.use(expressWinston.logger({
        transports: [
            new winston.transports.File({
            json: true,
            colorize: true,
            filename: './accessLogs'
              })
      ],
      meta: true,
      expressFormat: true,
 }));
```

C. Now access logs are generated in **/accesslog** file in JSON format given below.


```
{"res":{"statusCode":200},"req":{"url":"/login","headers":
{"host":"localhost:3011","connection":"keep-alive","content-
length":"82","user-agent":"Mozilla/5.0 (Macintosh; Intel Mac OS X
10_12_2) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/
66.0.3359.139 Safari/537.36","cache-control":"no-
cache","origin":"chrome-extension://
fhbjgbiflinjbdggehcddcbncdddomop","postman-token":"481a84b3-5297-
a340-bc91-f42ffc926348","content-type":"application/
json","accept":"*/*","accept-encoding":"gzip, deflate,
br","accept-language":"en-GB,en-
US;q=0.9,en;q=0.8"},"method":"POST","httpVersion":"1.1","originalU
rl":"/login","query":{},"body":
{"userName":"admin","password":"admin","userType":"admin","email":
""}},"responseTime":16,"level":"info","message":"POST /login 200
16ms","timestamp":"2018-05-23T21:05:31.007Z"}
```


D. You can store this accesslog file in any where in your instance.