

Node.js: how to read the nginx access log

In Node.js is quite easy to read the nginx access log.

Since Node is executed under a specific user account on a typical server, it can't access the nginx access log because it's located in the `/var/log/nginx` directory.

We need to copy the access log in the app directory and change its ownership accordingly.

```
cat /var/log/nginx/access.log >
/home/user/app/logs/access.log && chown user:user
/home/user/app/logs/access.log
```

Now we can create a cron job to automate the above task.

```
export VISUAL=nano; crontab -e
```

Typically this task can be done once in a day, as shown below.

```
5 4 * * * cat /var/log/nginx/access.log >
/home/user/app/logs/access.log && chown user:user
/home/user/app/logs/access.log
```

Then we can restart the cron service.

```
service cron restart
```

You can use this online service for your cron jobs. I recommend to run the above task not too frequently in order to minimize the impact on the server.

In Node we need to read the log file, split it into an array of lines and return the resulting array in reversed order since nginx stores its records in ascending order.

```
'use strict';

const fs = require('fs');

const readLog = () => {
  const logFile = './logs/access.log';
  return new Promise((resolve, reject) => {
    fs.readFile(logFile, 'utf8', (err, contents)
=> {
      if(err) { reject(err); }

      resolve(contents.toString().split(/\n/).reverse());
    });
  });
};

module.exports = readLog;
```

Now we can use the above function in our application:

```
'use strict';

const readLog = require('./readlog');
const app = require('express')();

app.set('view engine', 'ejs');
```

```
app.get('/log', async (req, res) => {
  try {
    const entries = await readLog();
    res.render('log', {entries: entries});
  } catch(err) { res.sendStatus(500); }
});

app.listen(8080);
```

Our EJS view may look like the following one:

```
<% if(entries.length > 0) { %>
<ol>
  <% entries.forEach(function(entry) { %>
    <li><%= entry %></li>
  <% }); %>
</ol>
<% } else { %>
  <p>No results to show.</p>
<% } %>
```

As you can see, once you get all the relevant data everything is really easy to accomplish.