

Node.js: minify CSS files in ExpressJS without using a module

In ExpressJS we can minify CSS files on the fly without using a module.

Minifying a CSS file simply means removing a set of characters that are ignored by the CSS parser (e.g. new lines, comments, unnecessary spaces etc.).

```
'use strict';

const app = require('express')();
const fs = require('fs');

const readCSS = path => {
  return new Promise((resolve, reject) => {
    if(!/\.css$/ .test(path)) {
      reject({error: 'Invalid file'});
    }
    if(!fs.existsSync(path)) {
      reject({error: 'File does not exist'});
    }

    fs.readFile(path, 'UTF-8', (err, contents) =>
    {
      if(err) {
        reject({error: err});
      } else {
        resolve(contents.toString());
      }
    }
  ));
}
```

```

    });
};

const minifyCSS = async (req, res) => {
  try {
    let cssFile = await readCSS('.' +
req.originalUrl);
    let buffer = cssFile.replace(/\s+/g, '
').replace(/\n|\r\n/g, '')
                                .replace(/\/\/*.*?\*\/g,
'')
                                .replace(/;\s+/g, ';')
                                .replace(/:\s+/g, ':')
                                .replace(/\s+\{/g, '{')
                                .replace(/\{\s+/g, '{')
                                .replace(/,\s+/g, ',')
                                .replace(/\}\s+/g, '}')
                                .replace(/;\}/g, '}');
    res.set('Content-Type', 'text/css');
    res.send(buffer);
  } catch(err) {
    res.sendStatus(404);
  }
};

app.get('/public/css/style.min.css', minifyCSS);

app.listen(4000);

```

In order to maximize performance you can also specify cache headers when serving the CSS file by using the ExpressJS `response.set()` method.