

```

/* Node.js static file web server */

// Importing necessary modules
const http = require('http');
const url = require('url');
const fs = require('fs');
const path = require('path');

// Port on which the server will create
const PORT = 1800;

// Maps file extension to MIME types which
// helps browser to understand what to do
// with the file
const mimeType = {
  '.ico': 'image/x-icon',
  '.html': 'text/html',
  '.js': 'text/javascript',
  '.json': 'application/json',
  '.css': 'text/css',
  '.png': 'image/png',
  '.jpg': 'image/jpeg',
  '.wav': 'audio/wav',
  '.mp3': 'audio/mpeg',
  '.svg': 'image/svg+xml',
  '.pdf': 'application/pdf',
  '.doc': 'application/msword',
  '.eot': 'application/vnd.ms-fontobject',
  '.ttf': 'application/font-sfnt'
};

// Creating a server and listening at port 1800
http.createServer( (req, res) => {

  // Parsing the requested URL
  const parsedUrl = url.parse(req.url);

  // If requested url is "/" like "http://localhost:1800/"
  if(parsedUrl.pathname==="/"){

```

```

var filesLink("<ul>");
res.setHeader('Content-type', 'text/html');
var fileList=fs.readdirSync("./");
fileList.forEach(element => {
    if(fs.statSync("./"+element).isFile()){
        filesLink+='\<br/><li><a href='./${element}'>
            ${element}
        </a></li>` ;
    }
});

filesLink+="</ul>";

res.end("<h1>List of files:</h1> " + filesLink);
}

/* Processing the requested file pathname to
avoid directory traversal like,
http://localhost:1800/../../fileOutOfContext.txt
by limiting to the current directory only. */
const sanitizePath =
path.normalize(parsedUrl.pathname).replace(/^(\\.\\. [\\/\]]+)/, '');

let pathname = path.join(__dirname, sanitizePath);

if(!fs.existsSync(pathname)) {

    // If the file is not found, return 404
    res.statusCode = 404;
    res.end(`File ${pathname} not found!`);
}
else {

    // Read file from file system limit to
    // the current directory only.
    fs.readFile(pathname, function(err, data) {
        if(err){
            res.statusCode = 500;
            res.end(`Error in getting the file.`);
        }
    }
}

```

```
else {

    // Based on the URL path, extract the
    // file extension. Ex .js, .doc, ...
    const ext = path.parse(pathname).ext;

    // If the file is found, set Content-type
    // and send data
    res.setHeader('Content-type',
        mimeType[ext] || 'text/plain' );

    res.end(data);
}

});

}

}).listen(PORT);

console.log(`Server listening on port ${PORT}`);
```

List of files:

- [app.js](#)
- [index.html](#)
- [package-lock.json](#)
- [package.json](#)