Autor: Mario Fentler
Datum: 13.11.2018

Webserver

Vorraussetzung

- Node.js installiert

Aufgabenstellung

Entwickle einen Webserver in node.js, der die zur im request angefragten url gehörende Datei aus dem lokalen Dateisystem laedt und an den Browser zurückgibt. Deine Loesung kann auf der vorigen Aufgabenstellung aufbauen, allerdings kein Framework wie express oder dergleichen verwendet werden. Achte auf eine sinnvolle Fehlerbehandlung (HTTP 404), falls die gesuchte Datei nicht existiert.

Erweiterung: Falls vom Client ein Verzeichnis angefragt wird, so liefere eine (verlinkte) Liste aller im Verzeichnis vorhandenen Dateien.

Aufgabendurchführung

Als erstes werden die notwendigen Konstanten geladen.

```
const http = require('http')
//URL Modul verwenden, da in der normalen URL Sonderzeichen enthalten sein können
const url = require('url')
const fs = require('fs')
```

Die Methode **"http.createServer((request,response) => {}).listen(8080)"** erstellt den Server und lässt ihn ihn diesem Beispiel auf den Port 8080 hören.

In dieser Methode wird die URL abgefragt und mit dem lokalen Pfad verbunden.

```
let requestedPath = url.parse(request.url).pathname
//process.cwd() gets me the current directory
let searchPath = process.cwd() + requestedPath
```

Anschließend wird dieser Pfad durchsucht und die Elemente (Files/Directories) werden weiterverwendet. Hier kommt nun eine **forEach** Funktion ins Spiel, die zu jeder Datei einen Link auf der Webseite erstellt.

```
http.createServer((request, response) => {
        let requestedPath = url.parse(request.url).pathname
        //process.cwd() gets me the current directory
        let searchPath = process.cwd() + requestedPath
```

```
        fs.readdir(searchPath, "utf-8", (err, allData) => {
                if(err) {
                fs.readFile(searchPath, "utf-8",(err,fileStr) =>{
                        if(err){
                                response.writeHead(404, {'Content-Type':
'text/html'})
                                response.write("File not found " + err)
                        }else{
                                response.writeHead(200, {'Content-Type':
'text/html'})
                                response.write('<h1>Content of file</h1>')
                                response.write(fileStr, function(err)
{response.end()})
                        }
                })
                }
                else{
                response.writeHead(200, {'Content-Type': 'text/html'})
                response.write('<h1>Required content of directory</h1>')

                allData.forEach(data => {
                        //Prints links to the webpage
                        response.write('<a href="http://localhost:8080'+
requestedPath +'/'+ data +'">' + data + '</a></br>', function(err)
{response.end()})
                })
                }
        })
}).
listen(8080)
```

WICHTIG: Die Funktion response.end() muss aufgerufen werden, damit die Webseite auf dem Firefox Browser angezeigt werden kann!

Directories lesen

Mit der Funktion *fs.readDir(path, (err,allData) =>{})* können Verzeichnisse ausgelesen werden. Sollte es sich beim Pfad um ein File handeln oder sollte sonst irgendein Fehler auftreten, dann wird die Error Methode aufgerufen. In dieser wird versucht das File mit der Methode *fs.readFile(path,(err,fileStr)=>{})* zu lesen.

```
fs.readdir(searchPath, "utf-8", (err, allData) => {
        if(err) {
        fs.readFile(searchPath, "utf-8",(err,fileStr) =>{
                ...
        })
        }
        else{
        response.writeHead(200, {'Content-Type': 'text/html'})
```

```
        response.write('<h1>Required content of directory</h1>')

        allData.forEach(data => {
                //Prints links to the webpage
                response.write('<a href="http://localhost:8080'+ requestedPath
+'/'+ data +'">' + data + '</a></br>', function(err){response.end()})
        })
        }
})
```

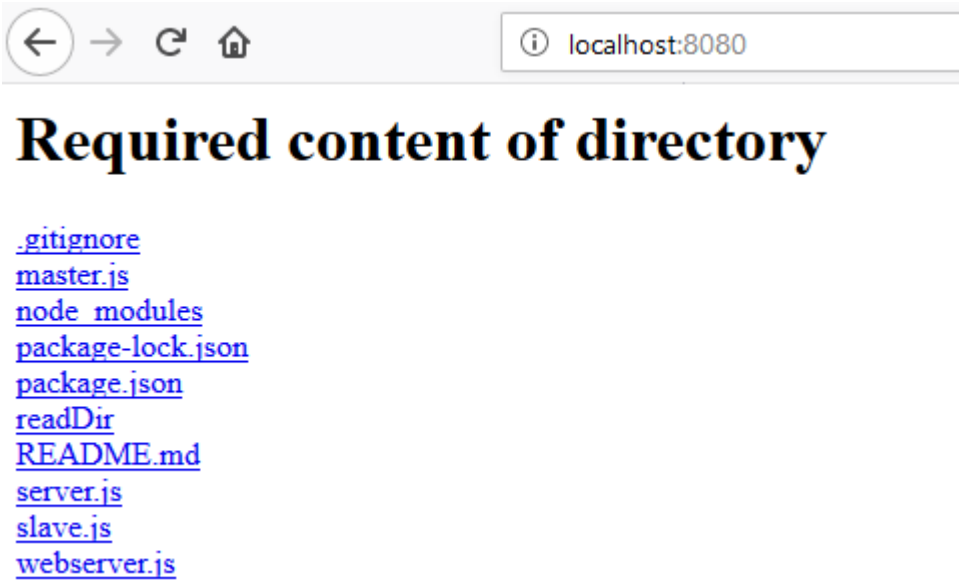
Files auslesen

Die Methode liest den Inhalt eines Files aus. Sollte es zu einem Error kommen wird dieser abgefangen und eine 404 HTTP Message ausgegeben.

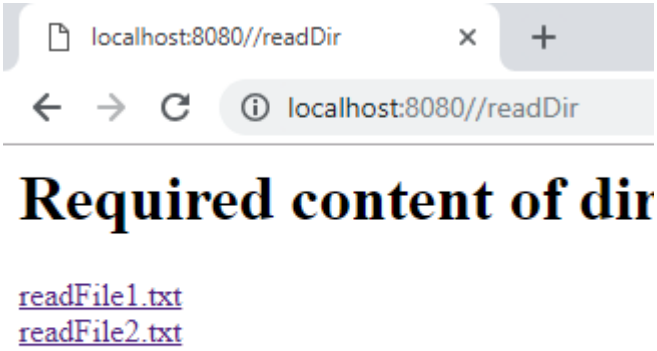
```
fs.readFile(searchPath, "utf-8",(err,fileStr) =>{
        if(err){
                response.writeHead(404, {'Content-Type': 'text/html'})
                response.write("File not found " + err)
        }else{
                response.writeHead(200, {'Content-Type': 'text/html'})
                //response.write('<h1>Content of file</h1>')
                //console.log(fileStr)
                response.write(fileStr, function(err){response.end()})
        }
})
```

Ergebnis

Nach dem man sich auf die Webseite "localhost:8080" verbunden hat, bekommt man alle Files und Directories aus dem Directory in dem das File liegt ausgegeben. Klickt man auf eines, so wird dieses angezeigt (siehe Bilder).



Wenn man dann auf einen Link klickt wird dieser geöffnet. Dabei wird überprüft ob es sich um ein Directory oder ein File handelt und dementsprechend gehandelt.



Wenn es ein File ist, dann wird hier der Inhalt ausgegeben.