NodeJs Fileserver
Luke Wood

Project Description

Node.js is currently one of the leading web development frameworks for creating dynamic web applications. Having a remote file-server is an incredibly handy tool, and so I decided to make one using Node.

Implementation

The first part of this project was to create a simple web-server that would allow the user to access any files that they needed, but to avoid cluttering the user's view with the code for the server. In order to do this I created two directories, one where the public files would stay and one where the server files would stay. I then created an interface that would allow the user to easily add, remove, and access files. Anytime a user uploads a file it is placed in the public directory and the file is added to the list of files on the homepage.

Code

The code for this project came in three pieces: the index.html page, the server-side javascript file, and the client-side javascript file. All three are displayed below.

Serverside Javascript

```
var port = 8000;
var serverUrl = "127.0.0.1";

var express = require("express");
var app = express();
var EventEmitter = require("events");
var http = require("http").Server(app);
```

```
var path = require("path");
var fs = require("fs");
var io =require("socket.io")(http);

app.use(express.static('priv'));
app.use(express.static('files'));

app.get("/",function(req,res)
{
        res.sendFile(__dirname +'\\index.html');
});

io.on("connection",function(socket)
{
        socket.on("requestfiles", function()
        {
                fs.readdir(__dirname + "/files", function(err,
files){socket.emit("filelist",files);});
        });

        socket.on("file upload", function(name, file)
        {
                console.log("File uploaded");
                fs.writeFile("files\\"+name,file, function(err){if(err){return
console.log(err);}});
                socket.emit("upload success", name);
                console.log("file should be saved");
        });
        socket.on("remove", function(name)
        {
                fs.unlink("files\\"+name);
                socket.emit("removeClientFile", name);
        });
});


http.listen(port, function()
{
                console.log("App Starting at " + serverUrl + ":" + port);
});
```

index.html

```
<html>
<head>
<script src="https://cdn.socket.io/socket.io-1.3.7.js"></script>

<script src = "external/dropzone.js"></script>
<link rel ="stylesheet" type = "text/css" href = "css/homepage.css">
<link rel="stylesheet" type="text/css" href="external/dropzone.css">
</head>
<body>
<center><h1>Luke's File Server</h1>
<h4>Upload Files:</h4>
<form class = "dropzone" action = "/upload-target">
<input type = "file" id = "fileSelector" name = "file" multiple size = "50"></input>
</form>
<button id = "upload">Upload</button>
<br>
Status: <div id = "status"></div>
<div id = "fileContainer">
<ul id = "files"></ul>
</div>
</center>
<script src = "clientjs/clientscript.js">
</script>


</body>
</html>
```

Clientside Javascript

```
var socket = io();
var uploadButton = document.getElementById("upload");
uploadButton.addEventListener("click",uploadButtonFunc);
var fileSelector = document.getElementById("fileSelector");
var statusDiv = document.getElementById("status");

var fileList = document.getElementById("files");

socket.on("upload success",function(name){
        statusDiv.innerHTML = "Files uploaded";
        fileSelector.value = fileSelector.defaultValue;
        setTimeout(
                function()
                {
                        statusDiv.innerHTML = "";
```

```
                }
                ,1000);
        (function()
                {
                var temp = document.createElement("li");
                temp.setAttribute("id",name);
                var removeBut = document.createElement("button");
                removeBut.innerHTML = "Remove";
                removeBut.addEventListener("click",function()
                {
                socket.emit("remove",name);
        });
        removeBut.setAttribute("style","float:left;");
        temp.appendChild(removeBut);

        var alink = document.createElement("a");
        alink.setAttribute("href",name);
        alink.innerHTML = name;
        temp.appendChild(alink);
        fileList.appendChild(temp);
        })();
});

socket.emit("requestfiles");

socket.on("filelist", function(fileNames)
{
        for(var i = 0; i < fileNames.length; i++)
        {
                (function()
                {
                        var a = i;
                        var temp = document.createElement("li");
                        temp.setAttribute("id",fileNames[a]);
                        var removeBut = document.createElement("button");
                        removeBut.innerHTML = "Remove";
                        removeBut.addEventListener("click",function()
                        {
                                socket.emit("remove",fileNames[a]);
                        });
                        removeBut.setAttribute("style","float:left;");
                        temp.appendChild(removeBut);

                        var alink = document.createElement("a");
                        alink.setAttribute("href",fileNames[a]);
                        alink.innerHTML = fileNames[a];
                        temp.appendChild(alink);
```

```
                    fileList.appendChild(temp);
            })();
        }
}
);

socket.on("removeClientFile",function(name)
{
        var el = document.getElementById(name);
        el.parentElement.removeChild(el);
        fileList.removeChild();
});

function uploadButtonFunc()
{
        if(fileSelector.files.length != 0)
        {
                for(var i = 0; i < fileSelector.files.length; i++)
                {
                        socket.emit("file upload",fileSelector.files[i].name,
fileSelector.files[i]);
                }
        }
        else
        {
                alert("Select one or more files");
        }
}
```

Results and Discussion

By using Node.js to complete this project, I was able to complete the task much more easily than had I been using php. One of the great things about Node.js is that it allows you to use Javascript for the entire project instead of being forced to change languages constantly. This allows for fast paced development of largely scalable applications.

Concluding Remarks

This project really got me hooked on Node.js. I plan on using it for something much larger in the near future. The power of being able to dynamically alter elements on the web page is both satisfying and rewarding.