Node JS Document

<https://nodejs.org/docs/latest-v17.x/api/>

<https://www.w3schools.com/nodejs/> (important link)

# File system Module

<https://www.w3schools.com/nodejs/ref_fs.asp>

To Play, Control, Read, Write, Delete, Update, Rename File Through JavaScript We Use File System

There are two types of file system in node JS modules

1. Synchronous file system (let me done and then run)
2. Asynchronous file system (don’t wait do your work I will be there )
3. **Synchronous file system**

* No call backs are allowed

const fs = require("fs")

//1..............synchronous file system

//create a folder and name it adil younas

fs.mkdirSync("./adil younas")

//create a file in it named bio.txt and add data into it.

fs.writeFileSync("./adil younas/bio.txt","hello i am adil younas and i am a full stack web developer")

//add more data in it (not override)

fs.appendFileSync("./adil younas/bio.txt"," asdjlfsdasdfjkl")

// read data without buffering

const a = fs.readFileSync("./adil younas/bio.txt","utf-8")

console.log(a);

// rename the folder name name to adil younas to aqil younas

fs.renameSync("./adil younas","aqil younas")

// rename the file in folder and named it mybio.txt

fs.renameSync("./aqil younas/bio.txt" , "./aqil younas/mybio.txt" )

// delete file from folder

fs.rmSync("./aqil younas/mybio.txt")

// delete folder

fs.rmdirSync("./aqil younas")

1. **Asynchronous file system**

* Asynchronous file system is all about call back

const fs = require("fs")

//2..............Asynchronous file system

//1..create a folder and name it adil younas

fs.mkdirSync("adil younas",(err)=>{

    console.log("folder created");

    console.log(err);

})

//2....create a file in it named bio.txt and add data into it.

fs.writeFile("./adil younas/bio.txt","my name is adil younas",(err)=>{

    console.log(err);

    console.log("file is created");

})

//3.....add more data in it (not override)

fs.appendFile("./adil younas/bio.txt","i am aqil younas",(err)=>{

    console.log("file updated");

    console.log(err);

})

//4..... read data without buffering

fs.readFile("./adil younas/bio.txt","utf-8",(err,data)=>{

    console.log(data);

    console.log(err);

})

//5...... rename the folder name name to adil younas to aqil younas

fs.rename("./adil younas","./abeeha younas",(err)=>{

    console.log("folder renamed");

    console.log(err);

})

// 6.... rename the file in folder and named it mybio.txt

fs.rename("./abeeha younas/bio.txt","./abeeha younas/mybio.txt",(err)=>{

    console.log("file renamed");

    console.log(err);

})

// 7... delete file from folder

fs.rm("./abeeha younas/mybio.txt",(err)=>{

    console.log(err);

    console.log("file deleted");

})

// 8..... delete folder

fs.rmdir("./abeeha younas",(err)=>{

    console.log(err);

    console.log("delete folder");

})

OS Modules

<https://www.w3schools.com/nodejs/ref_os.asp>

* Tell us All about you computer window ,memory, hostname, etc

Path Modules

<https://www.w3schools.com/nodejs/ref_path.asp>

* Tell us everything about file and path you can even join the paths

const path = require("path")

console.log(path.dirname("C:/Users/Aqil/Desktop/New folder/index.js"));

console.log(path.extname("C:/Users/Aqil/Desktop/New folder/index.js"));

console.log(path.basename("C:/Users/Aqil/Desktop/New folder/index.js"));

console.log(path.parse("C:/Users/Aqil/Desktop/New folder/index.js") );

const path1 = "adil younas"

const path2 = "/aqil younas"

console.log( path.join(path1 + path2)  );

wrapper function

* In node all data in a single file wrap in a function which is called wraper function

//considered this function a whole file

function wrapper(require,exports,modules,\_\_filename,\_\_dirname){

//all data you write in file of .js like

const fs = require("fs")

}

# **How to create a server**

const http = require("http");

const server = http.createServer( (req,res)=>{

    res.end("welcome to home page")

});

server.listen(8000, "127.0.0.1", ()=>{

    console.log("server is listening to the port 800")

});

# **Routing | Handled HTTP Requests**

const http = require("http");

const server = http.createServer( (req,res)=>{

    if(req.url== "/"){

        res.end("welcome to home page")

    }else if (req.url =="/about"){

        res.end("welcome to the about page")

    }else if (req.url =="/contact us"){

        res.end("welcome to the contact us page")

    }else if (req.url =="/our services"){

        res.end("welcome to our best services")

    }else{

        res.writeHead(404, {"Content-type" : "text/html"})

        res.end("<h1>404 error the page does't exist</h1>")

    }

});

server.listen(8000, "127.0.0.1", ()=>{

    console.log("server is listening to the port 800")

});

# **JSON File Format**

const Obj = {

    name:"adil younas",

    city:"Gujranwala",

    phone:"03040053938"

};

console.log(Obj)

//object to json

// Obj.name Obj.city can't be use in json formate

const jsonData = JSON.stringify(Obj)

console.log(jsonData)

//json to object

const orgObj = JSON.parse(jsonData)

console.log(orgObj)

# **Event Module in Node JS**

//first you create object

// onclick what happed is in event module like addEventListner in javaScript

const EventEmitter = require("events");

const event = new EventEmitter();

//can use two parameters in ()

event.on("sayMyName", ()=>{

    console.log("your name is Adil younas");

});

//to call event in node JS we use

// in () we can also use 2 arguments

event.emit("sayMyName");

//////////////////////////////////////////////////////////////////////

//we can make various events in node JS and call them at once like

const EventEmitter = require("events");

const event = new EventEmitter();

event.on("sayMyName", ()=>{

    console.log("your name is Adil younas");

});

event.on("sayMyName", ()=>{

    console.log("your name is Adil younas");

});

event.on("sayMyName", ()=>{

    console.log("your name is Adil younas");

});

event.emit("sayMyName");

//Console.log data

PS E:\html practice or portfolio\javascript projects\loops practice\event module> node eventModule.js

your name is Adil younas

your name is Adil younas

your name is Adil younas

PS E:\html practice or portfolio\javascript projects\loops practice\event module>

# **Streaming and Buffering In Node JS**

const fs = require("fs");

const http = require("http");

const server = http.createServer();

//res.write() is use to start data

//res.end() is use to end data

//both can use in streaming but in other file we can't use single item

//res.write() + res.end() use in other type

server.on("request", (req, res) => {

    const rstream = fs.createReadStream("inputs.txt");

    rstream.on("data", (chunkdata) => {

        res.end(chunkdata)

    });

    rstream.on("end", () => {

        res.end()

    });

    rstream.on("error", (err) => {

        res.end(err)

    })

});

server.listen(8000, "127.0.0.1");

# **Express Static Website Vs Dynamic Websites**

* Plzz For Further Details Of Folder Structure And More Check The Express JS Folder

//./Denotes the folder containing the html,css file

// \_\_dirname for the root folder

//This formula is for static websites

var express = require("express");

var app = express();

var path = require("path");

console.log(path.join(\_\_dirname, "./new folder"))

const staticPath = path.join(path.join(\_\_dirname, "./new folder"))

app.use(express.static(staticPath))

app.get( "/" , (req, res)=>{

    res.send("it home page")

});

app.listen(8000, ()=>{

    console.log("server is working")

})

# **Better Static and Dynamic Website Using Express JS with Project (reference from 100% dynamic website + weather API page)**

Main express .JS

//when i copy files (index.html data to index.hbs) to templates css folder remains in public folder in order to use it i need app.use(express.static(staticPath));

//the better option is that you first create a static website and after convert it into dynamic website

//always use one css file because of partial links options you create links of index links nav footer script etc. After that if you use further css links than its become headache

//event.preventDefault() is use to prevent page load again after submitng form i.g

// const getInfo = (event)=>{

//     event.preventDefault()

//     alert("lol")

// }

// plz make sure you define any

// innerHtml = ` ` always use bakticks if not = valiable

const express = require('express');

const app = express();

const hbs =require('hbs')

const path = require('path')

const port = process.env.PORT || 8000 //while hosting server give it port || use 8000 instead

// console.log(path.join(\_\_dirname, "../public"));  //for testing

const staticPath = path.join(\_\_dirname, "../public");

const template\_Path = path.join(\_\_dirname, "../templates/views");

const partials\_Path = path.join(\_\_dirname, "../templates/partials");

app.set('view engine', 'hbs');

app.set("views", template\_Path);

hbs.registerPartials(partials\_Path)

app.use(express.static(staticPath));

app.get("/index",(req,res)=>{

    res.render("index")

});

app.get("/about",(req,res)=>{

    res.render("about")

});

app.get("/weather",(req,res)=>{

    res.render("weather")

});

app.get("\*",(req,res)=>{

    res.render("404")

});

app.listen(port, ()=>{

    console.log(`server is listning at ${port}`)

});

Main JavaScript Page

let inputTextEl = document.querySelector('#input-text');

let submitBtnEl = document.getElementById('submitBtn');

let getOutPutEl = document.getElementById('getoutputhere');

let inputVal= "" //plz define a variable empty and than use it in function plz

// date , time , city name , degree , image........................

let dayEl = document.querySelector('.day');

let dateEl = document.querySelector('.date');

let monthEl = ""

let locationEl = document.querySelector('.location');

let countryEl = document.querySelector('.country');

let gujranwalaEl = document.querySelector('.gujranwala');

let cloudEl = document.querySelector('.cloud');

let tempEl = document.querySelector('.Tempa');

// const getInfo = (event)=>{

//     event.preventDefault()

//     getOutPutEl.innerText = inputVal

//     console.log(getOutPutEl)

// }

///////////////////////////////////////////////////////////

const basicInfo =()=>{

    const weekday = ["Sunday","Monday","Tuesday","Wednesday","Thursday","Friday","Saturday"];

    const month = ["January","February","March","April","May","June","July","August","September","October","November","December"];

    let date = new Date()

    monthEl = month[date.getMonth()]  //october

    dayEl.innerText = weekday[date.getDay()] //sunday

    console.log(monthEl)

    let twelve = date.getDate();

    dateEl.innerText = `${twelve} | ${monthEl}`

}

basicInfo()

//////////////////////////////////////////////

const getInfo = async(event)=>{

    event.preventDefault()

    inputVal = inputTextEl.value;

    if(inputVal === ""){

        getOutPutEl.innerText = "Enter City Name"

    }else{

        try{

            let url = `https://api.openweathermap.org/data/2.5/weather?q=${inputVal}&units=metric&appid=23023689f754c758f5fdb16bd8fb4041`

            let responseUrl = await fetch(url);

            let jsonData = await responseUrl.json();

            let temperatureVal = await jsonData.main.temp;

            tempEl.innerText = temperatureVal

            console.log(temperatureVal);

            let locationVal = await jsonData.name;

            gujranwalaEl.innerText = `${locationVal} , `

            console.log(locationVal);

            let countryVal = await jsonData.sys.country;

            countryEl.innerText = countryVal

            console.log(countryVal);

            let cloudVal = await jsonData.weather[0].main;

            if(cloudVal == "Clear"){

                // cloudEl.innerHTML = `<i class="fa-solid fa-sun-bright" style="color: #fff;"></i>`

                cloudEl.innerHTML = `Clear`

            }else if(cloudVal == "Rain"){

                cloudEl.innerHTML = `Rain`

            }else if(cloudVal == "Clouds"){

                cloudEl.innerHTML = `Clouds`

            }else if(cloudVal == "Sun"){

                cloudEl.innerHTML = `Sun`

            }else{

                cloudEl.innerHTML = `Clear`

            }

        }catch{

            getOutPutEl.innerText = `Plz Enter Valid City Name`;

        }

    }

}

submitBtnEl.addEventListener('click',getInfo);