API - Application programing interface

Simply API is data about a certain topic, We can call that api and use that data in our website.

API is like a contract between data provider and developer.

API allows us to interact with external server.

Endpoints, Paths, Parameters, Authentication

Endpoints: - This will go to the suffix of the url

Paths :- URL

Parameters: - To get specific data, it will go after endpoints in url

EX: - url/endpoints?parameterName=value&parameterName=value

Authentication: - To monetize or limit your use of data EX: - If a developer hits 60 request per minute its free, but if web app is deployed and used by 1000s of people, the data provider will keep track of the number of requests and will charge you accordingly

Postman is used for testing APIs.

JSON: - Javascript object notation

HTTPS status codes

200 - ok

404 - data not found

401 - unauthorized(mistake in api key)

To do a get request from our server to external server install https module using npm.

Do https.get(api_url, function(external response) { handle external response })

EX: -

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

```
})


app.post("/", function(req,res) {
    const query = req.body.cityName;
    const apiKey = "7be661621bd1b79439fc2c635d4a6391";
    const unit = "metric";
    var url = "https://api.openweathermap.org/data/2.5/weather?q=" + query + "&appid=" + apiKey + "&units=" + unit;
    https.get(url, function(externalResponse) {
        console.log(externalResponse.statusCode);
        externalResponse.on("data", function(data) {
            const weatherData = JSON.parse(data);
            const fetchedDataDescription = weatherData.weather[0].description;
            const fetchedDataTemp = weatherData.main.temp;
            const fetchedIcon = weatherData.weather[0].icon;
            const iconURL = "https://openweathermap.org/img/wn/" + fetchedIcon + "@2x.png";

            res.write("<h1> The temprature in " + req.body.cityName + " is " + fetchedDataTemp + " degree celcius" + "</h1>");
            res.write("<h3>" + fetchedDataDescription + "</h3>")
            res.write("<img src=" + iconURL + ">");
            res.send();
        })
    })
})
```

JSON.parse(data from external api response) - this will turn data in to javascript object.

JSON.stringify(A javascript object) - this will turn a javascript object into a string format.

To make our browser identify the local css files or local image :

app.use(express.static("public"));

- For above line to work, Make a folder called public and put css files or image in that

Code of newsletter mail

```
// Required modules
const express = require("express");
const bodyParser = require("body-parser");
const request = require("request");
const https = require("https");

const app = express();
app.use(bodyParser.urlencoded({extended: true}));
//This is for supporting local css files or image, must put in public folder
app.use(express.static("public"));

// Routes of application
app.get("/", function(req,res) {
   res.sendFile(__dirname + "/index.html");
})
app.post("/failure", function(req,res) {
   res.redirect("/");
})

// What Browser send us(data that user entered in form)
```

```
app.post("/", function(req,res) {
    console.log(req.body.firstName + " " + req.body.lastName + " " + req.body.email);

    const object = {
        members: [
            {
                email_address: req.body.email,
                status: "subscribed",
                merge_fields: {
                    FNAME: req.body.firstName,
                    LNAME: req.body.lastName,
                }
            }
        ]
    };
    const jsonData = JSON.stringify(object);
    const url = "https://us21.api.mailchimp.com/3.0/lists/5b214c97bd";
    const options = {
        method: "POST",
        auth: "chintan:8c022f6dbacb1227d8f3c9249abfd91c-us21",
    };

    // Based on what browser sent us, we make request to chimpmail api
    const requestToChimpmail = https.request(url,options,function(mailchimpResponse) {
        if(mailchimpResponse.statusCode == 200) { res.sendFile(__dirname + "/success.html"); } else {
```

```
res.sendFile(__dirname + "/failure.html"); }

        mailchimpResponse.on("data",function(data) {

            console.log(JSON.parse(data));

        })

    })

    requestToChimpmail.write(jsonData);

    requestToChimpmail.end();


})


// To listen to port number 3000

app.listen(3000, function() {

    console.log("Server started at port 3000");

})




// Api key

// 8c022f6dbacb1227d8f3c9249abfd91c-us21


// list id

// 5b214c97bd
```