**MERN Stack | Batch 1 - Day 2 Overview**

**Day 2 Agenda | Total Duration-2:16:04**

* API
* Creating basic API
* URL Parsing

**JSON :**

JSON is a stringified version of JS object.

We can create object which looks exactly like its object and we can write in key pairs

For multiple key values use square around curly braces.

**API:**

Application Programming Interface.This means a file which has code.

**WEB API:**

A Program present on the web  which has a lot of functions and based on URL we call the function and get the data in form of JSON or XML Format.

**Types of API:**

1. SOAP API(Simple Object Access Protocol)
2. REST API(Representational State Transfer)

**How REST API Works:**

If we have a server and inside it we have a file which has code which has 3 functions.

First Function: getUserData which will go to database collect all data and return as XML or JSON

Second Function: getSingleUser based on ID which will return all the data from user based on ID collected from the database and return as XML or JSON

Now these functions are connected to URL which is For Eg: [localhost:300](http://localhost:300/)

If I want to get all the data from the function.

So if we have the URL [localhost:300/users](http://localhost:300/users) which will call the function getUserData which will further call all the user data and get it from data base and return it in format of JSON or XML

For this we will have a client which is either browser or frontend program which will call the URL and get the details from above process and return to the client.

So this whole file from where the process is happening is in a browser so it is called a WEB API.

Import following module:

const http= require('http');

const fs=require('fs');

const url=require('url');

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

URL Parsing:

It request the url and Prints the details of the URL

const path=url.parse(req.url);

console.log(url.parse(req.url));

If the path is [localhost:300/](http://localhost:300/) or [localhost:300/products](http://localhost:300/products) then. it will read from the JSON file and print the products.

if(path.pathname=="/" || path.pathname=="/products"){

fs.readFile("./products.json","utf-8",(err,data)=>{

res.writeHead(200,{ //

"Content-type":"application/json"

})

res.end(data);

//if you want to print the file in console then un comment the statement written below

//console.log(data)

})

}

if path is [localhost:3000/product](http://localhost:3000/product) it will read data d=from JSON file and print the particular id specified

For Eg:If entered [localhost:3000/product?id=1](http://localhost:3000/product?id=1)

It will print the details of the product which has been given id 1

*NOTE: You need to specify the id of the product in you JSON file for this.*

else if(path.pathname=="/product"){

//console.log(path.query.split("="));

const id=path.query.split("=")[1];

fs.readFile("./products.json","utf-8",(err,data)=>{

const parseData=JSON.parse(data);

const singleData=parseData.filter((ele)=>{

return ele.id==id;

})

res.writeHead(200,{ //send 200 status

"Content-type":"application/json" //format to print data in JSON

})

res.end(JSON.stringify(singleData));//converting into string as data is object

})

//print below statement if you want to check the product URL is working fine

//res.end("This is gonna be single product");

}

If url dont exist we print Resourse not found

else{

res.writeHead(404,{

"Content-type":"text/html" //to accept the data in HTML

})//send 404 status

res.end("<h1>404 Resourse not found</h1>");

};

});

To start server and mention its properties

server.listen("3000","127.0.0.1",()=>{

console.log("Server is running");

});