Ndoe js concept

1. Difference between fs.wirtefilesync() and fs.writefile()

🡪 fs.wirtefilesync() is a synchronous . it does not require any call back to execute .

Where fs.writefile() is a asynchronous present of call back is must for the execution .

Note : we always try to use async : because it will be make our execution fast . so we always prefar to use fs.writefile() over fs.writefilesync() .

2. why do use ‘utf-8’ in fs.readfile() ?

It use to remove the buffer (a store binary value or space occupied during excuting node js file);

3. Creating a export module for multiple exports

*const* add=*function*(*a*,*b*){ return *a*+*b*;}

*const* sub=*function*(*a*,*b*){return *a*-*b* ;};

*const* multi=*function*(*a*,*b*){return *a*\**b*};

*module*.*exports*={add,sub,multi};

exporting file :

*const* {add,sub,multi}=require('./pr');

console.log(add(1,2));

console.log(sub(2,1));

console.log(multi(2,1));

4. chalk module

If I want my console ouput colourful then I can use chalk module .

<https://www.npmjs.com/package/chalk>

4.1 validation module :

Use for validation for various things

<https://www.npmjs.com/package/validator>

5. https status code

1. [Informational responses](https://developer.mozilla.org/en-US/docs/Web/HTTP/Status#information_responses) (100–199)
2. [Successful responses](https://developer.mozilla.org/en-US/docs/Web/HTTP/Status#successful_responses) (200–299)
3. [Redirection messages](https://developer.mozilla.org/en-US/docs/Web/HTTP/Status#redirection_messages) (300–399)
4. [Client error responses](https://developer.mozilla.org/en-US/docs/Web/HTTP/Status#client_error_responses) (400–499)
5. [Server error responses](https://developer.mozilla.org/en-US/docs/Web/HTTP/Status#server_error_responses) (500–599)

6. Routing in https

*const* http=require('http');

http.createServer((*reqs*,*resp*)*=>*{

    if(*reqs*.url=='/'){

*resp*.write("We are at the home page");

*resp*.end();

    }

    else if(*reqs*.url=='/about'){

*resp*.write("we are the About page");

*resp*.end();

    }

    else if(*reqs*.url=='/help'){

*resp*.write('we are at the help page');

*resp*.end();

    }

    else{

*resp*.writeHead(404,{'content-Type':'applicaton\jason'});

*resp*.write("404 bad error");

*resp*.end();

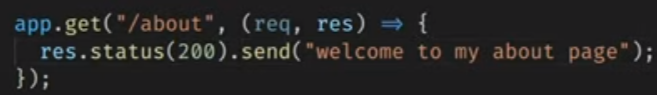
    }

}).listen(4500,()*=>*{

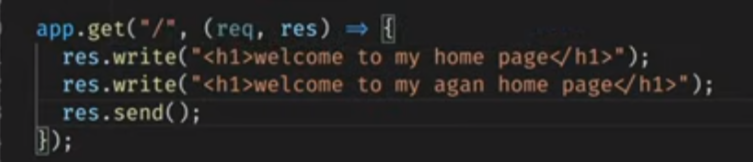
    console.log('server is running');

});

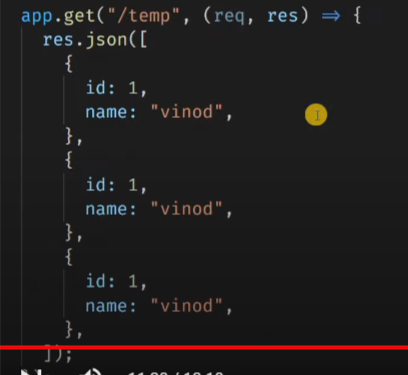
🡺 In express js we can give status code as well :



🡺 Writing multiple line of html In express js :



🡺 In express js we can send our response in resp.json() instead of resp.send():



Difference between resp.send() and resp.json():

1.

res.send uses content-type:text/html

res.json uses content-type:application/json

2.

The methods are identical when an object or array is passed, but res.json() will also convert non-objects, such as null and undefined, which are not valid JSON.

If your argument is not a JSON object or array (null, undefined, boolean, string), and you want to ensure it is sent as JSON, use res.json.