Using fs module

```
import fs from "fs"
fs.cop yFi le( " fil e1.t xt ", " fil e2.t xt " ,
(e rr) =>{
        if(err) consol e.l og(Error happened ${err
})
       else consol e.l og( " Copied Succes sfu -
lly.")
})
fs.wrt ieF ile ("me ssa ge.t xt ", " Hello from
Node Js!", (err) => {
if(err) consol e.log(Error happened ${err})
       else consol e.l og( "The file has been
saved." )
});
fs.rea dFi le( " mes sag e.t xt", " utf -8", (err,
data) => {
if(err) throw err;
       else consol e.l og( data);
});
```

node.js project initial setting

```
create folder (ex. intro_to_node)
npm init -y >> packag e.json created
npm i express nodemon
at packag e.json
" typ e": " mod ule ",
" mai n": " ser ver.js ",
" scr ipt s": {
...,
    " sta rt": " nodemon server.js "
}
create server.js file
```

Node/Express Server

```
import express from "express"
const app = express();
const PORT = 3000;
app.li ste n(P ORT ,() =>{
        con sol e(The server is up and running
on port ${PORT})
})
```

REST API

```
app.use(express.urlencoded({extended:true}))
//GET -- root route
app.ge t("/ " ,(r eq, res )=>{
       // consol e.l og( req.ra wHe aders)
        res.send(<h1>Hello </h 1>);
})
//GET -- about route
app.ge t("/ abo ut", (re q,r es) =>{
       // consol e.l og( req.ra wHe aders)
        res.js on( {"Na me": " Anu ", " Tit -
le": " Ins tru cto r"});
})
//POST
app.po st( " /", (req,r es) =>{
        res.se ndS tat us(201)
})
//PUT
app.pu t("/ ", (req,r es) =>{
        res.se nd( " Updated Succes sfu -
lly " ).s end Sta tus (200);
})
//DELETE
app.de let e("/ ", (req,r es) =>{
        res.se nd( " Deleted Succes sfu -
lly " ).s end Sta tus (200);
})
```

Ejs (Backend)

```
npm i ejs
import path from 'path';
import {fileU RLT oPath } from 'url';
const __filename = fileUR LTo Pat h(i mpo rt.m -
et a.url);
const __dirname = path.d irn ame (__ fil ename);
  //ejs
app.se t("view engine ", " ejs ");
app.se t("v iew s", path.j oin (__ dir name, " -
vie ws"));
//POST
app.po st( " /",( req ,re s)= >{
```


Sponsored by Readable.com
Measure your website readability!
https://readable.com

Ejs (Backend) (cont)

```
>    let {num1,num2}=req.body;
   let sum = parseInt(num1) + parseInt(num2);
   res.render("index",{data:sum, err:"There is an error."})
})
```

Ejs (Frontend)

```
<!DOCTYPE html>
<html lang="e n">
<he ad>
    <meta charse t="u tf- 8">
    <ti tle >Ca lcu lat or< /ti tle>
</h ead>
<bo dy>
    <h1 >Ca lcu lat or< /h1>
    <!-- action - URL or filename or path of server
side processing script -->
    <!-- action - it indicates where to send the
form inform ation when the form is submitted -->
    <!-- <method - post - transmits the form data
in the body of the http request -->
    <form action ="/" method ="po st">
        <label for="nu m1"> Enter first number :
</ lab el>
        <input type="n umb er" name="n um1 "
id=" num 1" placeh old er= " First Number "> <br >
<b r>
        <label for="nu m2"> Enter second number :
</ lab el>
        <input type="n umb er" name="n um2 "
id=" num 2" placeh old er= " Second Number ">
<br ><b r>
        <!-- <button type="s ubm it" name="s ubm -
it"> Cal cul ate </b utt on> -->
        <bu tto n>C alc ula te< /bu tto n>
    </f orm>
    <div <% if(loc als.data) {%>>
        <h3 >The sum is <%= data %>< /h3>
        <% } %>
    </d iv>

</b ody>
</h tml>
```

Axios

```
import axios from "axios"
app.po st( " /",async (req,r es) =>{
        const {categ ory ,di ffi culty} = req.body;
        const URL= https: //o pen tdb.co m/a pi.p hp ?a
dif fic ult y=$ {di ffi cul ty} &t ype =bo olean;
       try{
                const {data} = await axios.g et (URL)
                res.re nde r("i nde x",{ res Obj :da

       }
        cat ch( err){
                con sole.log( is ${err.m es sage});
                res.re nde r("i nde x", {error_db: e
       }
})
```

Fetch

```
//GET
app.ge t("/ " ,async (req,r es) =>{
        const URL= https: //c atf act.ni nja /fac
t;
       try{
//using fetch
                const response = await fetch( -
URL);
                const data = await respon se.j -
son();
                res.re nde r("i nde x", {data:
data})
       }
        cat ch( err){
                con sole.log( is ${err.m
 es sage});
                res.re nde r("i nde x", {error:
err.me ssage})
       }
})
```


Sponsored by Readable.com
Measure your website readability!
https://readable.com