

### Using fs module

```
import fs from "fs"
fs.copyFile("file1.txt", "file2.txt", (err) =>{
  if(err) console.log("Error happened ${err}")
  else console.log("Copied Successfully.")
})
fs.writeFile("message.txt", "Hello from Node.js!", (err) => {
  if(err) console.log("Error happened ${err}")
  else console.log("The file has been saved.")
});
fs.readFile("message.txt", "utf-8", (err, data) => {
  if(err) throw err;
  else console.log(data);
});
```

### node.js project initial setting

```
create folder (ex. intro_to_node)
npm init -y >> package.json created
npm install express nodemon
at package.json
"type": "module",
"main": "server.js",
"scripts": {
  ...,
  "start": "nodemon server.js"
}
create server.js file
```

### Node/Express Server

```
import express from "express"
const app = express();
const PORT = 3000;
app.listen(PORT, () =>{
  console.log("Server is up and running on port ${PORT}")
})
```

### REST API

```
app.use(express.urlencoded({extended:true}))
//GET -- root route
app.get("/", (req, res) =>{
  // console.log(req.headers)
  res.send("<h1>Hello </h1>");
})
//GET -- about route
app.get("/about", (req, res) =>{
  // console.log(req.headers)
  res.json({name: "Anu", title: "Instructor"});
})
//POST
app.post("/", (req, res) =>{
  res.sendStatus(201)
})
//PUT
app.put("/", (req, res) =>{
  res.send("Updated Successfully")
})
//DELETE
app.delete("/", (req, res) =>{
  res.sendStatus(200);
})
```

### Ejs (Backend)

```
npm install ejs
import path from 'path';
import {fileURLToPath} from 'url';
const __filename = fileURLToPath(import.meta.url);
const __dirname = path.dirname(__filename);
//ejs
app.set("view engine", "ejs");
app.set("views", path.join(__dirname, "views"));
//POST
app.post("/", (req, res) =>{
```

sally sung

By **sally sung**  
[cheatography.com/sally-sung/](https://cheatography.com/sally-sung/)

Not published yet.  
 Last updated 25th March, 2026.  
 Page 1 of 2.

Sponsored by **ApolloPad.com**  
 Everyone has a novel in them. Finish Yours!  
<https://apollopad.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="en">
<head>
  <meta charset="utf-8">
  <title>Calculator</title>
</head>
<body>
  <h1>Calculator</h1>
  <!-- action - URL or filename or path of server
side processing script -->
  <!-- action - it indicates where to send the
form information when the form is submitted -->
  <!-- <method - post - transmits the form data
in the body of the http request -->
  <form action="/" method="post">
    <label for="num1"> Enter first number :
</label>
    <input type="number" name="num1"
id="num1" placeholder=" First Number "> <br >
<br >
    <label for="num2"> Enter second number :
</label>
    <input type="number" name="num2"
id="num2" placeholder=" Second Number ">
<br ><br >
    <!-- <button type="submit" name="submit" -
it"> Calculate </button -->
    <button type="submit"> Calculate </button >
  </form>
  <div <% if(locals.data) {%>>
    <h3>The sum is <%= data %></h3>
    <% } %>
  </div>
</body>
</html>
```

### Axios

```
import axios from "axios"
app.post("/", async (req, res) =>{
  const {category, difficulty} = req.body;
  const URL = `https://open.tdb.com/api.php?adifficulty=${difficulty}&type=boolean`;
  try{
    const {data} = await axios.get(URL);
    res.render("index",{resObj: data});
  } catch(err){
    console.log(err);
    res.render("index", {error_db: err});
  }
})
```

### Fetch

```
//GET
app.get("/", async (req, res) =>{
  const URL = `https://catfact.ninja/fact`;
  try{
    //using fetch
    const response = await fetch(URL);
    const data = await response.json();
    res.render("index", {data: data});
  } catch(err){
    console.log(err);
    res.render("index", {error: err.message});
  }
})
```