

7. NPM & Tools

1. Install Material Icons
2. `npm init`
3. `npm Scripts`
4. `npm Packages`
5. Installing Packages
6. Installing nodemon
7. Using nodemon





7.1 Install Material Icons



Material Icon Theme v5.11.1

Philipp Kief

 25,491,060

★★★★★ (345)

♥ Sponsor

Material Design Icons for Visual Studio Code

Set File Icon Theme

Disable 

Uninstall 



Auto Update





React

7.2 npm init



npm init

```
prashantjain@Prashants-Mac-mini user % npm init
```

This utility will walk you through creating a package.json file.

It only covers the most common items, and tries to guess sensible defaults.

See `npm help init` for definitive documentation on these fields and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and save it as a dependency in the package.json file.

Press ^C at any time to quit.

package name: (user) User Backend

Sorry, name can only contain URL-friendly characters and name can no longer contain capital letters.

package name: (user) user-backend

version: (1.0.0)

description: This project will have the backend code of our User project.

entry point: (user.js) app.js

test command:

git repository:

keywords:

author: Complete Coding

license: (ISC)

About to write to /Users/prashantjain/workspace/NodeJS_Complete_YouTube/Chapter 6 - Event Loop/user/package.json:



React

7.3 npm Scripts



```
{
```

```
  "name": "user-backend",
```

```
  "version": "1.0.0",
```

```
  "description": "This project will have the backend code of our User  
project.",
```

```
  "main": "app.js",
```

▷ Debug

```
  "scripts": {
```

```
    "test": "echo \"Error: no test specified\" && exit 1",
```

```
    "start": "node app.js",
```

```
    "khul-ja-sim-sim": "node app.js"
```

```
  },
```

```
  "author": "Complete Coding",
```

```
  "license": "ISC"
```

```
}
```

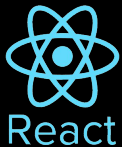
npm start

npm run khul-ja-sim-sim

7.4 npm Packages

1. **npm**: **Node.js** **package manager** for code sharing.
2. **Package**: **Reusable code** or library.
3. **package.json**: **Defines package metadata** and dependencies.
4. **Versioning**: Manages different package versions.
5. **Local/Global**: Install packages **locally** or **globally**.
6. **Registry**: **Public storage** for open-source packages.
7. **Examples**: **Express**, **React**, **Lodash**.





7.5 Installing Packages

`npm install <package-name>`

1. `-save`: Adds the package to the project's dependencies in `package.json`.
2. `-save-dev`: Adds the package to the project's `devDependencies` (used only in development) in `package.json`.
3. `-g`: Installs the package globally, making it available system-wide, not just in a specific project.
4. `-save-exact`: Installs the exact version specified without updating for newer versions.
5. `-force`: Forces npm to fetch and install packages even if they are already installed.



7.6 Installing nodemon



```
npm install nodemon --save-dev
```

user

node_modules

JS app.js

package-lock.json

package.json

JS user.js

user.txt

```
"scripts": {  
  "test": "echo \"Error: no test specified\" && exit 1",  
  "start": "node app.js",  
  "khul-ja-sim-sim": "node app.js"  
},  
"author": "Complete Coding",  
"license": "ISC",  
"devDependencies": {  
  "nodemon": "^3.1.7"  
}
```

npm install

Recreates node_modules



React

7.7 Using nodemon



```
"scripts": {  
  "test": "echo \"Error: no test specified\" && exit 1",  
  "start": "nodemon app.js",  
  "khuḷ-ja-sim-sim": "node app.js"  
},
```

```
prashantjain@Prashants-Mac-mini user % nodemon app.js  
zsh: command not found: nodemon  
prashantjain@Prashants-Mac-mini user % npm start
```

`npm install nodemon -g`

```
> user-backend@1.0.0 start  
> nodemon app.js
```

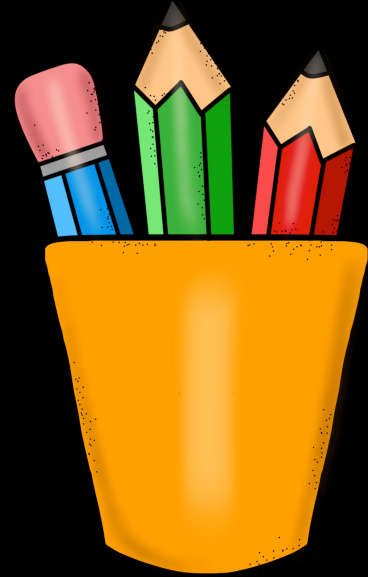
```
[nodemon] 3.1.7  
[nodemon] to restart at any time, enter `rs`  
[nodemon] watching path(s): *.*  
[nodemon] watching extensions: js,mjs,cjs,json  
[nodemon] starting `node app.js`  
Server running on address http://localhost:3001
```




React

Revision

1. Install Material Icons
2. `npm init`
3. `npm Scripts`
4. `npm Packages`
5. Installing Packages
6. Installing nodemon
7. Using nodemon





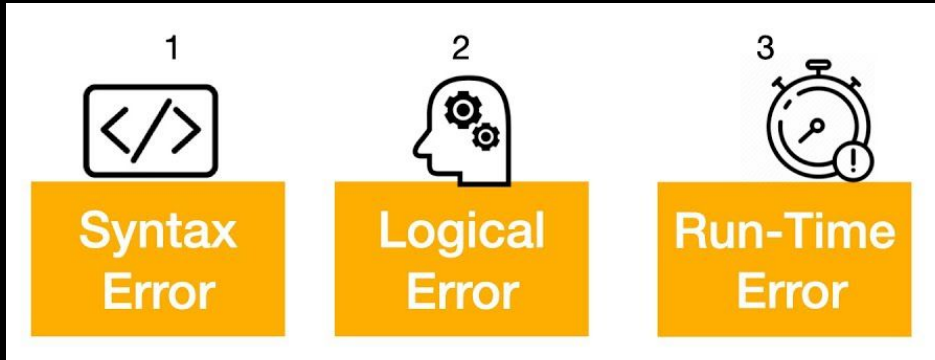


8. Errors & Debugging

1. Types of Errors
2. Syntax Errors
3. Runtime Errors
4. Logical Errors
5. Using the Debugger
6. Debugger with Async Code
7. Restart Debug with nodemon



8.1 Types of Errors



- 1. Syntax Error:** An **error** in the **code's structure**, causing it to not compile or run (e.g., missing semicolon).
- 2. Logical Error:** The code runs but **produces incorrect results** due to **faulty logic** (e.g., wrong formula).
- 3. Runtime Error:** An **error** that occurs while the program is running, often due to **invalid operations**



React

8.2 Syntax Errors

```
// Missing parenthesis in function call
```

```
console.log("Hello, world"
```

```
// Unclosed string literal
```

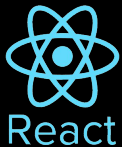
```
let message = "Welcome to Node.js; ~
```

```
// Improper use of reserved keywords
```

```
let new = 5;
```

```
// Incorrect variable declaration (const needs an initial value)
```

```
const myVar;
```



8.3 Runtime Errors

```
Error [ERR_HTTP_HEADERS_SENT]: Cannot set headers after they are sent to the client
    at ServerResponse.setHeader (node:_http_outgoing:699:11)
    at IncomingMessage.<anonymous> (/Users/prashantjain/workspace/Test Project/node/app.js:44:11)
    at IncomingMessage.emit (node:events:532:35)
    at endReadableNT (node:internal/streams/readable:1696:12)
    at process.processTicksAndRejections (node:internal/process/task_queues:82:21) {
  code: 'ERR_HTTP_HEADERS_SENT'
}
```

```
// Reference Error (x is not defined)
```

```
console.log(x);
```

```
// Type Error (num is not a function)
```

```
let num = 10;
```

```
num();
```

```
// Invalid JSON parse (SyntaxError)
```

```
let jsonString = "{ name: 'John' }"; // Invalid JSON (single quotes)
```

```
JSON.parse(jsonString);
```

```
// File not found error (fs module)
```

```
const fs = require('fs');
```

```
fs.readFileSync('nonexistentFile.txt'); // Throws Error: ENOENT (file not found)
```



React

8.4 Logical Errors

```
let x = 5;  
if (x = 10) { // Assignment instead of comparison  
  console.log("x is 10"); // Incorrectly prints this  
}
```

```
let arr = [1, 2, 3, 4, 5];  
for (let i = 0; i <= arr.length; i++) {  
  console.log(arr[i]); // Prints undefined at the end of the loop  
}
```

```
let num = "10";  
console.log(num + 5); // Expected result: 15, prints 105
```



React

8.5 Using the Debugger

Step 1

Go Run Terminal Window Help

Start Debugging

Run Without Debugging

Step 2

Select debugger

Node.js

VS Code Extension Development

Web App (Chrome)

Web App (Edge)

Install an extension for JavaScript...

Step 3: Put a breakpoint

```
1 let x = 5;
2 if (x = 10) { // Assignment instead of comparison
3   console.log("x is 10"); // Incorrectly prints this
4 }
5
```




8.5 Using the Debugger

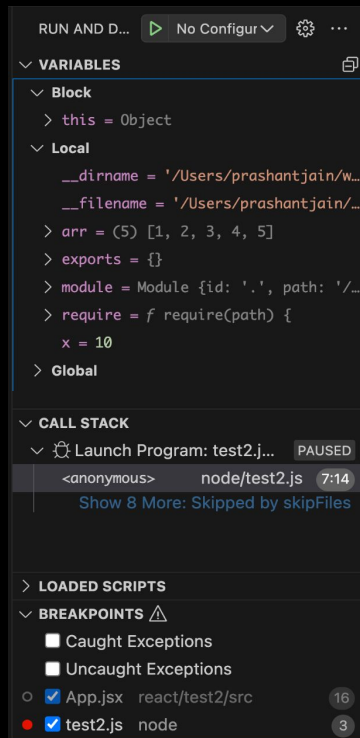
Step 4: Use the tools



Step 5: Hover



Step 6: Debug Panel





8.5 Using the Debugger

Step 7: Using Debug Console

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

```
/opt/homebrew/bin/node ./node/test2.js
```

```
→ x
  10
→ x++
  10
→ x === 15
  false
```

```
>
```



React

8.6 Debugger with Async Code

```
• 31 console.log("Here");
32 req.on("end", () => {
• 33   const parsedBody = Buffer.concat(body).toString();
34   console.log(parsedBody);
35   const params = new URLSearchParams(parsedBody);
36   const jsonObject = {};
37   for (const [key, value] of params.entries()) {
38     jsonObject[key] = value;
39   }
40   const jsonString = JSON.stringify(jsonObject);
41   console.log(jsonString);
42   // Async Operation
43   fs.writeFile("user-details.txt", jsonString, error => {
• 44     res.setHeader("Location", "/");
45     res.statusCode = 302;
46     return res.end();
47   });
48 });
49 }
```



8.7 Restart Debug with nodemon



Run Terminal Window

Start Debugging

Run Without Debugging

Stop Debugging

Restart Debugging

Open Configurations

Add Configuration...

Step Over

Step Into

```
"version": "0.2.0",
"configurations": [
  {
    "type": "node",
    "request": "launch",
    "name": "Launch Program",
    "skipFiles": [
      "<node_internals>/**"
    ],
    "program": "${workspaceFolder}/node/test2.js",
    "restart": true,
    "runtimeExecutable": "nodemon",
    "console": "integratedTerminal"
  }
]
```



React

Practise Set

Debug and fix Syntax, Runtime and Logical Errors

```
function calculateArea(width, height {  
  return width + height;  
}
```

```
let width = 10 height = 5;
```

```
if (area > 100) {  
  console.log("The area is large.");  
} else {  
  console.log("The area is small.");  
}
```

```
if (width + height > 100) {  
  console.log("Area is greater than or equal to 100");  
}
```





Revision

1. Types of Errors
2. Syntax Errors
3. Runtime Errors
4. Logical Errors
5. Using the Debugger
6. Debugger with Async Code
7. Restart Debug with nodemon

