

Can be told to rpm while installing:

```
npm install <package_name> --save-dev [Dev]
```

-- save [7 mod]  
→ -g [globally]

-- save [7 mod]  
→ -g [globally]

In the configuration file (package.json), we have a scripts section.

special script- none

Now we can execute

npm start in console.

npm start - server (Error)

↳ Not a known command.

custom script name  $\rightarrow$  npm run  $\langle$ script\_name $\rangle$

### Installing 3<sup>rd</sup> Party Packages:

Core Node Packages  
(fs / http / os)

cloud package repository  
that store all these  
packages.

прт керосину

Often, this does not suffice.

experiments

body - pattern

managed via ept

## Dependence

(3rd Party)

Some functionality (code) we didn't write on our own, but also not included in Node.js.

ex → Router helps with parsing requests, validating user-input etc.

`npm install nodemon`

↳ auto restart on save

dependencies  
nodemon: "1.18.3" <sup>has</sup> package will be updated if we re-run  
(Pick latest version while reinstalling)

package-lock.json → stores exact version we installed on that day.

Saathi

Date \_\_\_/\_\_\_/\_\_\_

nodemon →

Utility tool that allows us to run our application, which runs node app.js in the backend, but also watches our files for changes, and restarts the process.

Now, "start" → "nodemon app.js"

looks for nodemon tool, which it finds in project (as we've installed it).

Note:

If we run nodemon app.js in console, we get error: command not found.

→ only installed in project, not on the machine.

(Terminated twice to find it globally).

⊗ "npm scripts will look for LOCAL version of package"

Logical Error →

most difficult to fix but we can ~~use~~ use debugger.

→ not give error message.

Parsing body to receive message →

```
const message = parsedbody.split("=")[0];
```

Node.js debugger has a great integration into VS Code.

app.js file → Debug → start debugging

(Choose Node.js environment)

→ we can look into the code as it's running.

to do this, we need to set some breakpoints.

once we run, we get highlighted yellow line around our breakpoint.

→ Code execution has stopped on this line.

(Analyze code right in the moment while it's running).

View → Debug

gives a special debug mode where we see key variables available right now.

debug → add configuration (Node.js)

"console": "integratedTerminal"

"restart": true, (→ global install reqd).

"runtimeExecutable": "nodemon"

Page No.