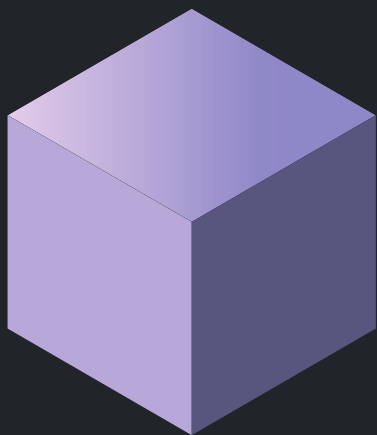




Save it



Like it



npm

# package.json & package-lock.json



@coder\_aishya



You can initialize the node project by running the below command

```
npm init
```

As you initialize your node application, you will see three files installed in your app that is `node_modules`, `package.json`, and `package.lock.json`.



@coder\_aishya



# package.json

- package.json is a versioning file used to install multiple packages in your project.
- It contains metadata about the project and also the functional dependencies that is required by the application.

```
{  
  "name": "Your project name",  
  "version": "1.0.0",  
  "description": "Your project description",  
  "main": "app.js",  
  "scripts": {  
    "test": "echo \"Error: no test specified\" && exit 1",  
  },  
  "author": "Author name",  
  "license": "ISC",  
  "dependencies": {  
    "dependency1": "^1.4.0",  
    "dependency2": "^1.5.2"  
  }  
}
```



- package.json is mandatory for every project
- Application name/version/scripts(ng scripts)

## main packages sections

### dependencies

- packages mandatory for production
- rxjs/core/http/bootstrap/jquery
- minified application size

### dev dependencies

- packages mandatory for development
- cli/testing (karma/jasmine)
- package-version number

These packages may not be installed in our project but may be needed .



@coder\_aishya



we can see installed packages in  
package.lock.json

## package.lock.json

- This file is automatically generated for those operations where npm modifies either the node\_module tree or package.json
- It is generated after npm install
- It allows future devs & automated systems to download the same dependencies as the project
- It also allows you to go back to the past version of dependencies without actual committing the node\_modules folder



## ●●● package.lock.json

```
{
  "name": "Your project name",
  "version": "1.0.0",
  "lockfileVersion": 1,
  "requires": true,
  "dependencies": {
    "dependency1": {
      "version": "1.4.0",
      "resolved":
"https://registry.npmjs.org/dependency1/-/dependency1-
1.4.0.tgz",
      "integrity": "sha512-
a+UqTh4kgZg/SlGvfbzDHpgRu7AAQ0mmqRHJnxhRZICKFUT91brVhNNt58CM
WU9PsBbv3PDCZUHbVxuDiH2mtA=="
    },
    "dependency2": {
      "version": "1.5.2",
      "resolved":
"https://registry.npmjs.org/dependency2/-/dependency2-
1.5.2.tgz",
      "integrity": "sha512-
W0n21V8AhyE1QqVfPIVxe3tupJacq1xGkPTB4iagT6o+P2cAgE00wIxMftr4
+ZCTI6d551ij9j61DFr0nsP2uQ=="
    }
  }
}
```



# Why is package-lock.json created?

```
npm i <package-name> -- save
```

it will install the exact latest version of that package in your project and save the dependency in package.json with a caret (^) sign.

## Example

if the current version of a package is 5.2.3 then the installed version will be 5.2.3 and the saved dependency will be ^5.2.3. Carat (^) means it will support any higher version with major version 5 like 5.3.1 and so on.



@coder\_aishya



## What is the purpose or use of package-lock.json?

To avoid differences in installed dependencies on different environments and to generate the same results on every environment we should use the package-lock.json file to install dependencies.

## Why should we commit package-lock.json with our project source code?

if you wanted to have that particular version for your dependency during deployment which you used at the time of development. This is the need of creating a **package-lock.json** file and keeping it with the source code.





**Follow me for more**



aishwarya-dhuri



coder\_aishya