

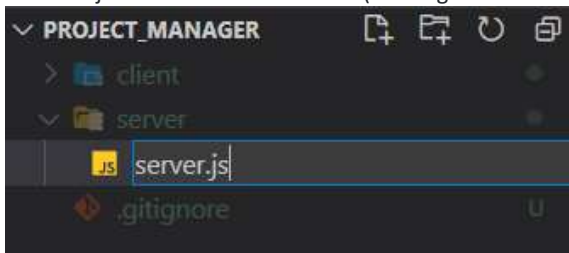
# MERN Full Stack Project Creation Cheat Sheet

Thursday, May 21, 2020 3:54 PM

## MERN Basic Project Creation Cheat Sheet

REMOVE THE `{{TEXT}}` and replace the text with your own

1. open terminal to where you want your project to be created
2. `mkdir {{PROJECT_NAME}}`
3. `cd {{PROJECT_NAME}}`
4. Setup your project to use a git repository:  
`git init`
5. Setup a .gitignore file to avoid adding "node\_modules" for both server and client folders:  
`echo node_modules/ > .gitignore`
6. Commit your .gitignore file to ensure you don't accident
7. Create the client folder for your front-end react app  
`npx create-react-app client`
8. create server folder:  
`mkdir server`
9. Open VSCode to your `{{PROJECT_NAME}}` folder
10. Create "server.js" file in the server folder (nothing needs to be inside of it yet)



11. Open a VSCode terminal and go into your server folder  
`cd server`
12. Initialize a new back-end node project (your server) with a package.json  
`npm init -y`
13. install back-end dependencies  
`npm install express mongoose cors`

If you plan to include socket.io install the following:

```
npm install socket.io
```

To create a login and registration based on the platform, be sure to install the dotenv package:

```
npm install dotenv
```

**\*\* Your server's package.json should now show have these dependencies listed**

14. in VSCode terminal: open a 2nd VSCode terminal (split terminal icon or plus icon) and navigate to the client folder  
`cd client`

15. install front-end dependencies, usually

```
npm install axios @reach/router
```

If you get an error stating: " ERESOLVE unable to resolve dependency tree" you can re-run the command and add " --legacy-peer-deps" to the end

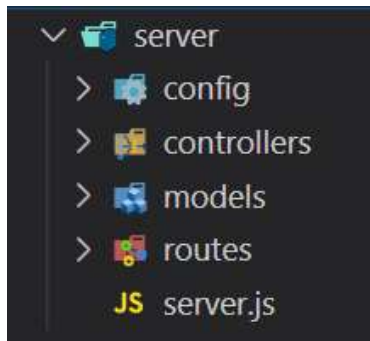
ex. `npm install axios @reach/router --legacy-peer-deps`

If you plan to include socket.io install the following:

```
npm install socket.io-client
```

Your react app's package.json (inside client folder) should now have these dependencies listed

16. create the modularized back-end folder structure and files in the server folder following the learn platform or lecture video / lecture code



For initial file setup see the Full Stack MERN section of the platform  
<http://learn.codingdojo.com/m/146/6960/50361>

17. After you have created your backend files and functionality, run your backend server and test from the server folder in your first terminal, run the server:

```
nodemon server.js
```

\*\* If you have followed along, you should already have one terminal open to the client folder and one in the server folder

18. Test all back-end CRUD functionality using postman

\*\* Include checking for mongo IDs, invalid lengths or missing keys that are required (schema validation) with post, put and delete

19. After you have built out the front end functionality, run the react server and test from client folder start your react app:

```
npm start
```

\*\* Make sure your server is still running!

\*\* If you fully tested your backend server with postman, any errors found will be with your frontend code

---

The following is a great setup for a full stack MERN `/project_name/.gitignore` file  
(It is much more complete than what you start with in this cheat sheet)

# See <https://help.github.com/articles/ignoring-files/> for more about ignoring files.

# dependencies

```
node_modules/  
.pnp/  
.pnp.js
```

# testing

```
coverage/
```

# production

```
build/
```

# misc

```
.vscode/  
.DS_Store  
.env  
.env.local  
.env.development.local  
.env.test.local  
.env.production.local
```

```
npm-debug.log*
```

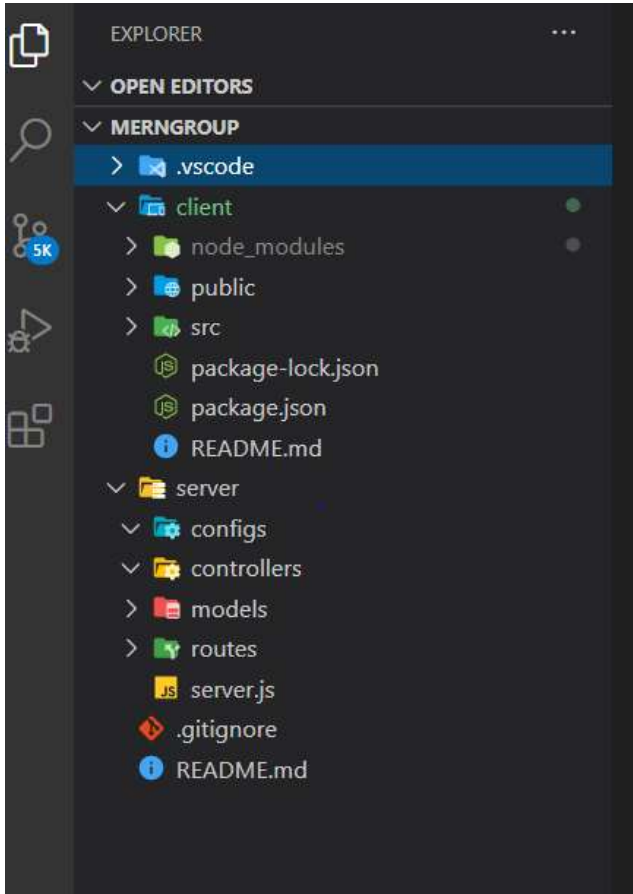
```
yarn-debug.log*
```

```
yarn-error.log*
```

---

## Completed project file start

\*\* You may or may not have a .vscode folder



## Troubleshooting:

```
$ npm install axios @reach/router
npm ERR! code ERESOLVE
npm ERR! ERESOLVE unable to resolve dependency tree
npm ERR!
npm ERR! While resolving: client@0.1.0
npm ERR! Found: react@17.0.2
npm ERR! node_modules/react
npm ERR!   react@"^17.0.2" from the root project
npm ERR!
npm ERR! Could not resolve dependency:
npm ERR! peer react@"15.x || 16.x || 16.4.0-alpha.0911da3" from @reach/router@1.3.4
npm ERR! node_modules/@reach/router
npm ERR!   @reach/router@"*" from the root project
npm ERR!
npm ERR! Fix the upstream dependency conflict, or retry
npm ERR! this command with --force, or --legacy-peer-deps
npm ERR! to accept an incorrect (and potentially broken) dependency resolution.
npm ERR!
```

retry the command and add --legacy-peer-deps

ex. `npm install axios @reach/router --legacy-peer-deps`