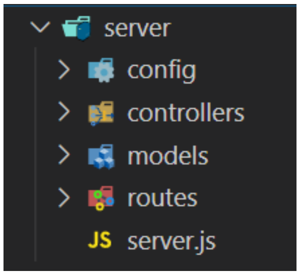
MERN Basic Project Creation Cheat Sheet

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. `npx create-react-app client` to create a client folder for your front-end react app
7. create server folder: `mkdir server`
8. `code .` to open VSCode to your `{{PROJECT_NAME}}` folder
9. create "server.js" file in the server folder (nothing needs to be inside of it yet)
10. Open a VSCode terminal
11. `cd server`
12. `npm init -y` to initialize a new back-end node project (your server) with a package.json
13. install back-end dependencies, usually `npm i express mongoose cors socket.io`
    - your back-end package.json should now show have these dependencies listed
14. in VSCode terminal: open a 2nd VSCode terminal (split terminal icon or plus icon)
15. `cd client`
16. install front-end dependencies, usually `npm i axios @reach/router socket.io-client`
    - your front-end package.json (inside client folder) should now have these dependencies listed
17. create the modularized back-end folder structure and files in the server folder following the learn platform or lecture video / lecture code



```
server
  config
  controllers
  models
  routes
  server.js
```

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

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 with `nodemon server.js`

- one VSCode terminal should be open to client folder and one should be open to server folder
- Test all back-end CRUD functionality using postman
    - including sending in invalid lengths or missing keys that are required

After you have built out the front end functionality, run the react server and test
from client folder start your react app with `npm start` and your node server with `nodemon server.js`

- you should now have two VSCode terminals open,
    - one open to client folder running the react app
    - one open to server folder running your server.js
- If you fully tested your backend server with postman, any errors found will be with your frontend code