






React + Node JS Project Setup

| | |
|---|--|
|  Author |  Jay Vachhani |
|  Category | JS |
|  Date Started | @Aug 24, 2020 |
|  Last Revision | @Aug 24, 2020 |

Acknowledgement:

I would like to personally thank all of the open source community in their continued effort to invent and innovate for the greater good of technology.

I have used multiple sources in writing this document. From online courses/articles/websites to also my own personal learning and knowledge. By no means am I claiming to be the sole creator of all of the information displayed here, It is a collection of curated material from myself and the wider web. I have written this document out myself, should you feel there are any major inconsistencies or inaccurate information being presented here, please contact me using the tab "Contact Me" on my website jayvachhani.tech , please use the name of this document in your message.

REACT + Node JS

This document is meant as reference for when you want to start a new React + Node JS project. The following instructions will setup a repository which when running will have your React App running on port 3001 and Express running on port 3000. Upon completion of the setup, when you run `npm install` and then `npm start` after cloning the repo both apps should start on your local system.

This is ideal for windows users, there may be some differences experienced by Linux/MacOS users however they will likely be minor.

It is recommended that you do this on VScode with its in built terminal as it will allow easy folder navigation.

Repository Setup

Create Repository on Github/Bitbucket

Create a repository on a platform of your choice and then clone it to your local system. It does not matter if it has any files in it or not before you clone it. Clone it using the link by typing the url after `git clone` on your terminal/bash.

```
git clone https://github.com/user/project.git
```

"project" will be used as a placeholder

React Setup

- `cd` into your project directory in your terminal
- Run the following commands one after the others' completion on the terminal in the directory
 - `npm i -g create-react-app` → installs the react app dependencies globally
 - `create-react-app client` → creates a client directory which is the frontend application
 - `cd client` → moves into client directory

- Go into the package.json and add `"proxy":`
`" http://localhost:3001/ ",` before the dependencies

This ensures that Express will be running on Port 3000 and they don't crash by running on the same port.

```
"proxy": "http://localhost:3001/",
"dependencies": {
  "react": "16.4.1",
  "react-dom": "16.4.1",
  "react-scripts": "1.1.4"
},
```

Node JS Setup (Express)

- `cd` into your project directory in your terminal, outside of `client` directory
- Run the following commands one after the others' completion on the terminal in the directory

`mkdir server` → creates a server directory which will run express

`npm install express-generator -g` → installs express dependencies globally

`express --view=ejs server` → sets up server directory as a node JS express project

`cd server` → moves inside the server directory

`npm install` → installs required packages

`npm install nodemon` → installs nodemon which keeps our server running and refreshes the app as we change and save it live.

- Open `package.json` and find the following code:

```
"scripts": {  
  "start": "node ./bin/www"  
}
```

```
"scripts": {  
  "start": "nodemon ./bin/www"  
}
```

Replace "node" with "nodemon"

- Navigate to `server/bin/www`
- Find a line stated as `var port = normalizePort(process.env.PORT || '3000');`
Change the port from `'3000'` to `'3001'` or any you like, except 3000

- `cd` back to the `project` directory outside client and server
- In the command line execute

`npm init -y` → this initialises the project directory as a node project

- Go to the `package.json` file which is outside both client and server directories

Add the `client` , `server` & `start` scripts to `scripts` :

```
"scripts": {  
  "client": "cd client && npm start",  
  "server": "cd server && npm start",  
  "start": "concurrently-kill-others \"npm run server\" \"npm run client\"",  
},
```

Also add the following after license (Don't forget the comma after license):

```
"license": "ISC",  
"devDependencies": {  
  "concurrently": "3.5.1"  
}
```

Conclusion

Now when you are in the project directory running `npm install` (also run this in both client and server directories) then `npm start` (run this outside both client and server in your project directory) should start your react app on your browser at address <http://localhost:3000> and your express app in the background running the backend. With nodemon, you won't need to continuously start/stop the app to check changes, you can start it once and then as you save the server restarts.

Ensure that when you commit/push changes that you add `.gitignore` files which should specify not to include node_modules in pushing, furthermore you should ensure both the `package.json` and `package.json.lock` are pushed to the repository as well.