

Connecting the Frontend (React) to backend (Node/Express)

Walkthrough guide by Dr Anand – NUS School of Computing

A. Install necessary tools:

Open your command prompt as ADMIN (right click on the command prompt icon/menu and open as administrator). For mac/linux users just open terminal.

Install React:

sudo npm -g install create-react-app [windows users remove 'sudo']

Install Express:

sudo npm install -g express-generator [windows users remove 'sudo']

Install CORS:

sudo npm install -g save cors [windows users remove 'sudo']

B. Create Project Folder:

Use Windows Explorer/Finder to navigate to your Desktop and create a folder 'fullstackapp'. This 'fullstackapp' will be your project folder.

C. Create backend and frontend projects:

Open your command prompt (right click on the command prompt icon/menu and open as administrator). For mac/linux users just open terminal.

Navigate to your project folder (using cd command)

Create Backend:

express --view=pug backend

Move to backend directory & install dependencies.

cd backend

npm install

Create Frontend:

Move one level up (cd ..) to 'fullstackapp' folder and type the following.

npx create-react-app frontend

D. Checking backend and frontend projects:

Open your command prompt (right click on the command prompt icon/menu and open as administrator). For mac/linux users just open terminal.

Navigate to your project folder (using cd command)

- 1) Start your backend to check whether it is working.

Navigate to '**backend**' folder and issue the following start command

npm start

Open browser and type **http://localhost:3000** you should be able to see 'Express' 'Welcome to express'.

Stop backend by pressing ctrl+c in the same window where you types 'npm start')

- 2) Start your frontend to check whether it is working.

Navigate to '**frontend**' folder and issue the following start command

npm start

Open browser and type **http://localhost:3001** you should be able to see 'React welcome page with react logo'

Stop backend by pressing ctrl+c in the same window where you types 'npm start')

Note: In-case you want to kill a process running in a specific port use,

lsof -i:3000 - to find the process id

kill -9 PID - to kill the process with process id= PID.

E. Editing the Backend in VScode

Open VS Code, open to our project folder "fullstackapp". Navigate to backend.

1. Open the file **/bin/www**

Change sever port to 7000. [The following lines should be already there, just change the port number].

```
var port = normalizePort(process.env.PORT || '7000');  
app.set('port', port);
```

2. Inside **router** folder,
Create a new file called '**mypage.js**'
Copy the following code to mypage.js and save.

```
var express = require('express'); //import
var router = express.Router();

var mystr = "I am sending this DATA from backend to Frontend";

/* GET my page. */
router.get('/', function(req, res, next) {
  res.send(mystr);
});

module.exports = router;
```

3. Open **app.js** and add the lines marked as “//Add THIS LINE – x” in the code below. There are FOUR “//Add THIS LINE – x” comments. [Alternatively, you can replace all the codes in your app.js with the following code].

```
var createError = require('http-errors');
var express = require('express');
var path = require('path');
var cookieParser = require('cookie-parser');
var logger = require('morgan');
var cors = require("cors"); //Add THIS LINE - 1

var indexRouter = require('./routes/index');
var usersRouter = require('./routes/users');
var mypageRouter = require('./routes/mypage'); //Add THIS LINE - 2

var app = express();

// view engine setup
app.set('views', path.join(__dirname, 'views'));
app.set('view engine', 'pug');

app.use(logger('dev'));
app.use(express.json());
app.use(express.urlencoded({ extended: false }));
app.use(cookieParser());
app.use(express.static(path.join(__dirname, 'public')));

app.use(cors()); //Add THIS LINE - 3
app.use('/', indexRouter);
app.use('/users', usersRouter);
app.use('/mypage', mypageRouter); //Add THIS LINE - 4

// catch 404 and forward to error handler
app.use(function(req, res, next) {
  next(createError(404));
});

// error handler
app.use(function(err, req, res, next) {
  // next handler not handling error in development
```

4. **Install cors locally.** [for some users, global installation in section A was not successful]

- Navigate to **backend folder** in your terminal/command window
- Type the following command Install CORS:
sudo npm install cors [windows users remove 'sudo']

If you still get cors error, try

npm install cors [windows users try in user mode, not in Admin mode for this]

[**Note:** The command is different from part A. There is no '-g' flag.]

5. As stated in section D above start your backend server and check whether it is working.

In the browser, type

http://localhost:7000/mypage

The browser should show:

I am sending this DATA from backend to Frontend

If you see this, Great Job!

<Continued in next page>

F. Editing the Frontend in VScode

Frontend and not backend

Open VS Code, open to our project folder “fullstackapp”. Navigate to backend.

1. Open /src/App.js, delete everything and copy the following content.

```
import React from 'react';
import logo from './logo.svg';
import './App.css';

class App extends React.Component {
  constructor(props) {
    super(props);
    this.state = { serverResponse: "" };
  }

  callAPIServer() {
    fetch("http://localhost:7000/mypage")
      .then(res => res.text())
      .then(res => this.setState({ serverResponse: res }))
      .catch(err => err);
  }

  componentDidMount() { // react lifecycle method componentDidMount()
    //will execute the callAPIServer() method after the component mounts.
    this.callAPIServer();
  }

  render() {
    return (
      <div className="App">
        <header className="App-header">
          <img src={logo} className="App-logo" alt="logo" />
          <h1 className="App-title">Welcome to FintechSG React Course</h1>

          <h2 className="App-intro">{this.state.serverResponse}</h2>
        </header>
      </div>
    );
  }
}
```

to start backend in cmd. Go to VS code, right click frontend folder and open in integrated terminal and start frontend from there.

2. As stated in section D above start your frontend server and check whether it is working.

In the browser, type

http://localhost:3000

The browser should show the following text below the React logo:

I am sending this DATA from backend to Frontend

If you see this, Great Job!

Congratulations! You have successfully received data from your BACKEND(express) to FRONTEND(react)

G. What is next?

Try to understand the reason for each of the above steps and the source code changes! Discuss with me if you need help!