Top of Form

Bottom of Form

[Donate](https://www.freecodecamp.org/donate/)

**Stay safe, friends. Learn to code from home.**[**Use our free 2,000 hour curriculum.**](https://www.freecodecamp.org/)

7 SEPTEMBER 2018/[#EXPRESS](https://www.freecodecamp.org/news/tag/express/)

**How to create a React frontend and a Node/Express backend and connect them**



by João Henrique

In this article, I’ll walk you through the process of creating a simple React app and connecting it to a simple Node/Express API that we will also be creating.

I won't go into much detail about how to work with any of the technologies I will mention in this tutorial but I will leave links, in case you want to learn more.

You can find all the code in [**this repository**](https://github.com/Joao-Henrique/React_Express_App_Medium_Tutorial) I made for the tutorial.

The objective here is to give you a practical guide on how to set up and connect the **front-end client** and the**back-end API**.

Before we get our hands dirty, make sure you have [Node.js](https://nodejs.org/en/) running on your machine.

**Create the Main Project directory**

In your terminal, navigate to a directory where you would like to save your project. Now create a new directory for your project and navigate into it:

mkdir my\_awesome\_project

cd my\_awesome\_project

**Create a**[**React**](https://reactjs.org/)**App**

This process is really straightforward.

I will be using Facebook’s [create-react-app](https://github.com/facebook/create-react-app) to… you guessed it, easily create a react app named **client**:

npx create-react-app client

cd client

npm start

*Let’s see what I have done:*

1. *Used npm’s*[*npx*](https://medium.com/@maybekatz/introducing-npx-an-npm-package-runner-55f7d4bd282b)*to create a react app and named it client.*
2. *cd(change directory) into the client directory.*
3. *Started the app.*

In your browser, navigate to <http://localhost:3000/>.

If all is ok, you will see the react welcome page. Congratulations! That means you now have a basic [**React**](https://reactjs.org/) application running on your local machine. Easy right?

To stop your react app, just press **Ctrl + c** in your terminal.

**Create an**[**Express**](https://expressjs.com/)**App**

Ok, this will be as straightforward as the previous example. Don’t forget to navigate to your project top folder.

I will be using the [Express Application Generator](https://expressjs.com/en/starter/generator.html) to quickly create an application skeleton and name it **api:**

npx express-generator api

cd api

npm install

npm start

*Let’s see what I have done:*

1. *Used npm’s npx to install express-generator globally.*
2. *Used express-generator to create an express app and named it api.*
3. *cd into the API directory.*
4. Installed all dependencies.
5. Started the app.

In your browser, navigate to <http://localhost:3000/>.

If all is ok, you will see the express welcome page. Congratulations! That means you now have a basic [**Express**](https://expressjs.com/) application running on your local machine. Easy right?

To stop your react app, just press **Ctrl + c** in your terminal.

**Configuring a new**[**route**](https://expressjs.com/en/guide/routing.html)**in the Express API**

Ok, let’s get our hands dirty. Time to open your favorite code editor *(I’m using*[*VS Code*](https://code.visualstudio.com/)*)*and navigate to your project folder.

*If you named the****react app as client****and the****express app as api****, you will find two main folders:****client****and****api.***

1. Inside the **API** directory, go to **bin/www** and change the port number on line 15 from 3000 to 9000. We will be running both apps at the same time later on, so doing this will avoid issues. The result should be something like this:

my\_awesome\_project/api/bin/www

2. On **api/routes**, create a **testAPI.js** file and paste this code:

var express = require(“express”);

var router = express.Router();

router.get(“/”, function(req, res, next) {

res.send(“API is working properly”);

});

module.exports = router;

3. On the **api/app.js** file, insert a new route on line 24:

app.use("/testAPI", testAPIRouter);

4. Ok, you are “telling” express to use this route but, you still have to require it. Let’s do that on line 9:

var testAPIRouter = require("./routes/testAPI");

The only changes are in line 9 and line 25. It should end up something like this:

my\_awesome\_project/api/app.js

5. Congratulations! You have created a new [route](https://expressjs.com/en/guide/routing.html).

If you start your API app (in your terminal, navigate to the API directory and “**npm start”**), and go to <http://localhost:9000/testAPI> in your browser, you will see the message: ***API is working properly.***

**Connecting the React Client to the Express API**

1. On your code editor, let’s work in the **client** directory. Open **app.js** file located in **my\_awesome\_project/client/app.js**.
2. Here I will use the [**Fetch API**](https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API/Using_Fetch)to retrieve data from the **API.**Just paste this code after the Class declaration and before the render method:

constructor(props) {

super(props);

this.state = { apiResponse: "" };

}

callAPI() {

fetch("http://localhost:9000/testAPI")

.then(res => res.text())

.then(res => this.setState({ apiResponse: res }));

}

componentWillMount() {

this.callAPI();

}

3. Inside the render method, you will find a **<**;p> tag. Let’s change it so that it render**s the apiRes**ponse:

<p className="App-intro">;{this.state.apiResponse}</p>

At the end, this file should look something like this:



I know!!! This was a bit too much. Copy paste is your friend, but you have to understand what you are doing. Let’s see what I did here:

1. *On lines 6 to 9, we inserted a constructor, that initializes the default state.*
2. *On lines 11 to 16, we inserted the method****callAPI()****that will fetch the data from the API and store the response on****this.state.apiResponse.***
3. *On lines 18 to 20, we inserted a react lifecycle method called****componentDidMount(),****that will execute the****callAPI()****method after the component mounts.*
4. Last, on line 29, I used the **<**;p> tag to display a paragraph on our client page, with the text that we retrieved from the API.

**What the heck!!**[**CORS**](https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS)**?**

Oh yeah, baby! We are almost done. But if we start both our apps (client and API) and navigate to <http://localhost:3000/>, you still won't find the expected result displayed on the page. If you open chrome developer tools, you will find why. In the console, you will see this error:

Failed to load <http://localhost:9000/testAPI>: No ‘Access-Control-Allow-Origin’ header is present on the requested resource. Origin ‘[http://localhost:3000'](http://localhost:3000'/) is therefore not allowed access. If an opaque response serves your needs, set the request’s mode to ‘no-cors’ to fetch the resource with CORS disabled.

This is simple to solve. We just have to add CORS to our API to allow cross-origin requests. Let’s do just that. You should [check here](https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS) to find out more about [CORS](https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS).

1. In your terminal navigate to the API directory and install the **CORS** package:

npm install --save cors

2. On your code editor go to the API directory and open the **my\_awesome\_project/api/app.js** file.

3. On line 6 require **CORS:**

var cors = require("cors");

4. Now on line 18 “tell” express to use **CORS**:

app.use(cors());

The API **app.js** file should end up something like this:

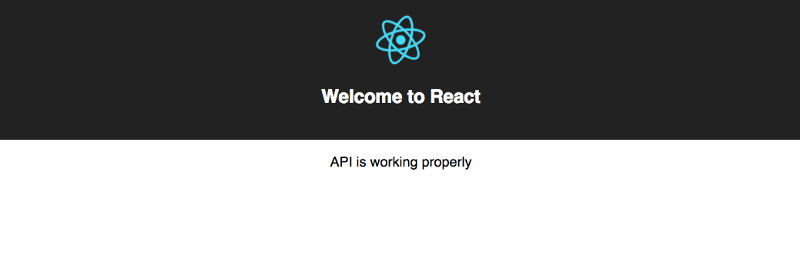
my\_awesome\_project/api/app.js

**Great Work. It’s all done!!**

Ok! We are all set!

Now start both your apps (client and API), in two different terminals, using the **npm start** command.

If you navigate to <http://localhost:3000/> in your browser you should find something like this:



Of course, this project as it is won’t do much, but is the start of a Full Stack Application. You can find all the code in [**this repository**](https://github.com/Joao-Henrique/React_Express_App_Medium_Tutorial) that I’ve created for the tutorial.

Next, I will work on some complementary tutorials, like how to connect this to a MongoDB database and even, how to run it all inside Docker containers.

As a good friend of mine says:

Be Strong and Code On!!!

…and don't forget to be awesome today ;)

If this article was helpful, tweet it.

Learn to code for free. freeCodeCamp's open source curriculum has helped more than 40,000 people get jobs as developers. [Get started](https://www.freecodecamp.org/learn)

freeCodeCamp is a donor-supported tax-exempt 501(c)(3) nonprofit organization (United States Federal Tax Identification Number: 82-0779546)

Our mission: to help people learn to code for free. We accomplish this by creating thousands of videos, articles, and interactive coding lessons - all freely available to the public. We also have thousands of freeCodeCamp study groups around the world.

Donations to freeCodeCamp go toward our education initiatives, and help pay for servers, services, and staff.

**You can**[**make a tax-deductible donation here**](https://freecodecamp.org/donate)**.**

**Trending Guides**

[JavaScript Closure](https://www.freecodecamp.org/news/javascript-closure-tutorial-with-js-closure-example-code/)[CSS Box Shadow](https://www.freecodecamp.org/news/css-tutorial-drop-shadow/)[Python List Append](https://www.freecodecamp.org/news/python-list-append-how-to-add-an-element-to-an-array-explained-with-examples/)[JavaScript Array Sort](https://www.freecodecamp.org/news/javascript-array-sort-tutorial-how-to-use-js-sort-methods-with-code-examples/)[Symlink in Linux](https://www.freecodecamp.org/news/symlink-tutorial-in-linux-how-to-create-and-remove-a-symbolic-link/)[Linux Grep Command](https://www.freecodecamp.org/news/grep-command-tutorial-how-to-search-for-a-file-in-linux-and-unix/)[What is DNS?](https://www.freecodecamp.org/news/what-is-dns/)[Primary Key SQL](https://www.freecodecamp.org/news/primary-key-sql-tutorial-how-to-define-a-primary-key-in-a-database/)[SQL Update Statement](https://www.freecodecamp.org/news/sql-update-statement-example-queries-for-updating-table-values/)[Screenshot on PC](https://www.freecodecamp.org/news/how-to-screenshot-on-windows/)

[JavaScript Promise](https://www.freecodecamp.org/news/javascript-es6-promises-for-beginners-resolve-reject-and-chaining-explained/)[What is GitHub?](https://www.freecodecamp.org/news/what-is-github-what-is-git-and-how-to-use-these-developer-tools/)[Python Sort List](https://www.freecodecamp.org/news/the-python-sort-list-array-method-ascending-and-descending-explained-with-examples/)[Comments in JSON](https://www.freecodecamp.org/news/json-comment-example-how-to-comment-in-json-files/)[What is Kanban?](https://www.freecodecamp.org/news/what-is-kanban-the-agile-methodology-defined-and-how-to-use-it-for-your-software-development-team-2/)[Python Write to File](https://www.freecodecamp.org/news/python-write-to-file-open-read-append-and-other-file-handling-functions-explained/)[CSS Media Queries](https://www.freecodecamp.org/news/css-media-queries-breakpoints-media-types-standard-resolutions-and-more/)[HTML Entities](https://www.freecodecamp.org/news/html-entities-symbols-special-character-codes-list/)[Excel VBA](https://www.freecodecamp.org/news/excel-vba-tutorial/)[LOOKUP in Excel](https://www.freecodecamp.org/news/vlookup-in-excel/)

[What is a Proxy Server?](https://www.freecodecamp.org/news/what-is-a-proxy-server-in-english-please/)[Cat Command in Linux](https://www.freecodecamp.org/news/the-cat-command-in-linux-concatenation-explained-with-bash-examples/)[CSS Background Image](https://www.freecodecamp.org/news/how-to-add-an-image-url-to-your-div/)[HTML Background Color](https://www.freecodecamp.org/news/html-background-color-tutorial-how-to-change-a-div-background-color-explained-with-code-examples/)[CSS Comment Example](https://www.freecodecamp.org/news/comments-in-css/)

[Arrow Function JavaScript](https://www.freecodecamp.org/news/arrow-function-javascript-tutorial-how-to-declare-a-js-function-with-the-new-es6-syntax/)[Remove Duplicates in Excel](https://www.freecodecamp.org/news/how-to-remove-duplicates-in-excel-delete-duplicate-rows-with-a-few-clicks/)[dllhost.exe COM Surrogate](https://www.freecodecamp.org/news/what-is-dllhost-exe-and-com-surrogate-in-windows-task-manager-solved/)[Boolean Algebra Truth Table](https://www.freecodecamp.org/news/boolean-algebra/)[Video Chat for Android](https://www.freecodecamp.org/news/can-you-facetime-on-android-no-but-here-are-some-alternative-video-conferencing-apps/)

[About](https://www.freecodecamp.org/news/about/)[Alumni Network](https://www.linkedin.com/school/free-code-camp/people/)[Open Source](https://github.com/freeCodeCamp/)[Shop](https://www.freecodecamp.org/shop/)[Support](https://www.freecodecamp.org/news/support/)[Sponsors](https://www.freecodecamp.org/news/sponsors/)[Academic Honesty](https://www.freecodecamp.org/news/academic-honesty-policy/)[Code of Conduct](https://www.freecodecamp.org/news/code-of-conduct/)[Privacy Policy](https://www.freecodecamp.org/news/privacy-policy/)[Terms of Service](https://www.freecodecamp.org/news/terms-of-service/)[Copyright Policy](https://www.freecodecamp.org/news/copyright-policy/)