

## Leonardo Bruno Lima

Follow

294 Followers About

# React tips — Working with relative path using create-react-app



Leonardo Bruno Lima Aug 3, 2018 · 4 min read

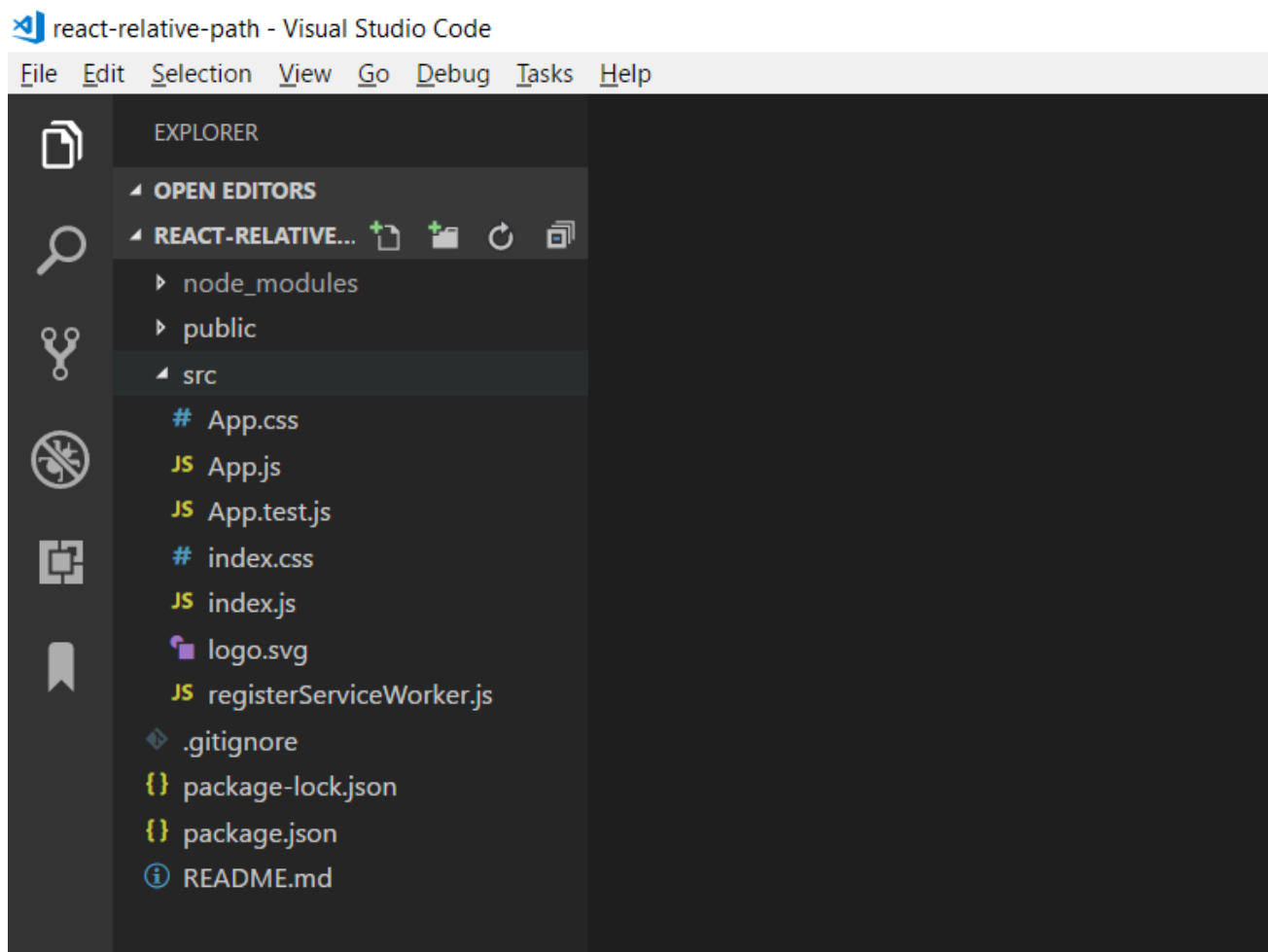


Photo by [Jack Anstey](#) on [Unsplash](#)

Deal with relative path is not a complex thing, but when we are creating an application using **create-react-app** sometimes it could be a tough work. To configure paths and

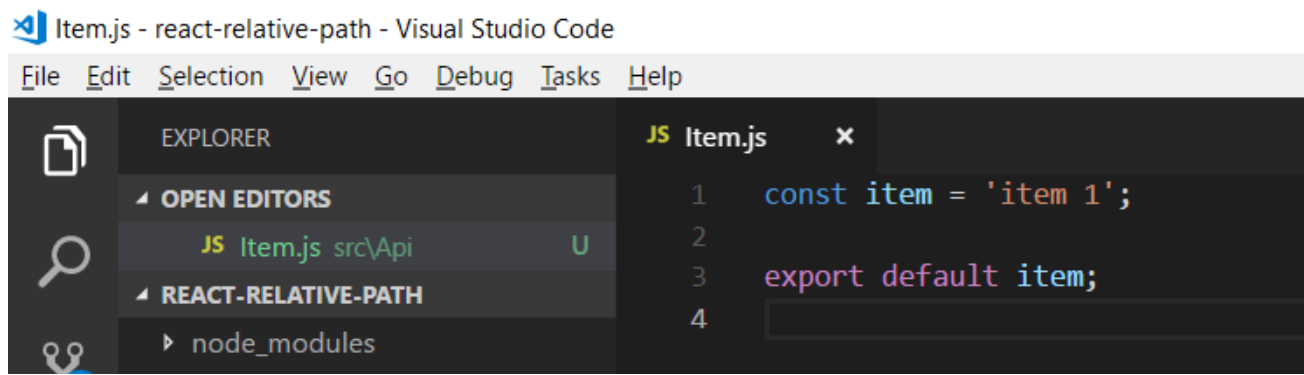
In this post I will show you how to change path configuration using Babel without eject your application. Let's create an app using create-react-app called react-relative-path:

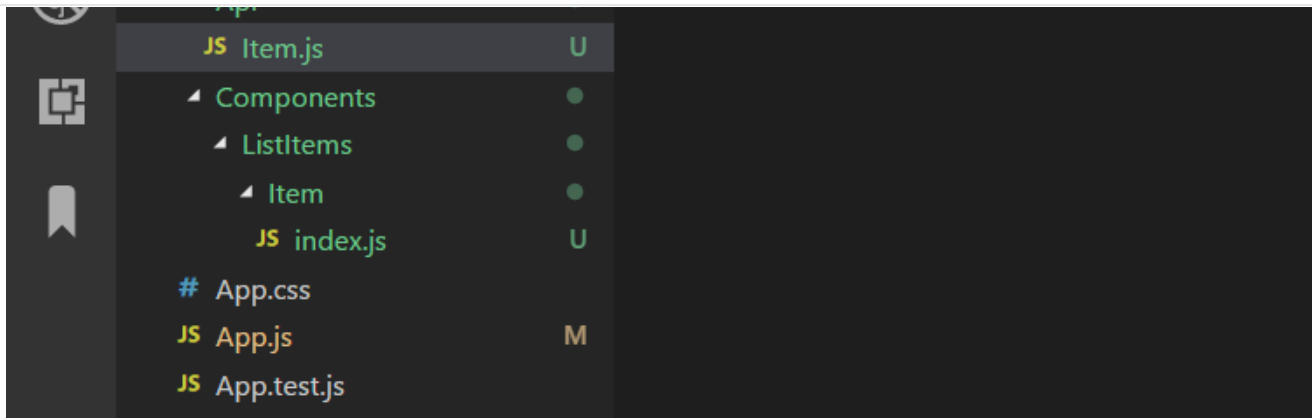
```
$ create-react-app react-relative-path
```



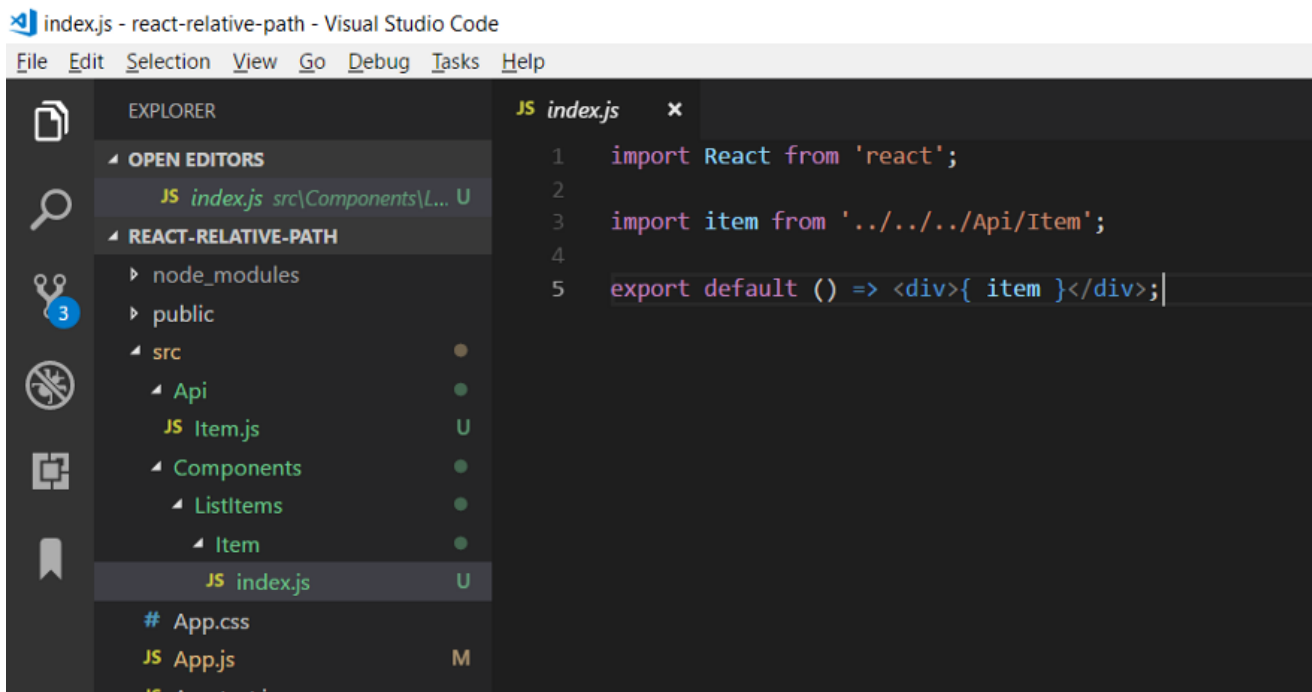
Now, I will create some folders, add some components and add references to each other.

First create a folder called **Api** and add a Item.js file.

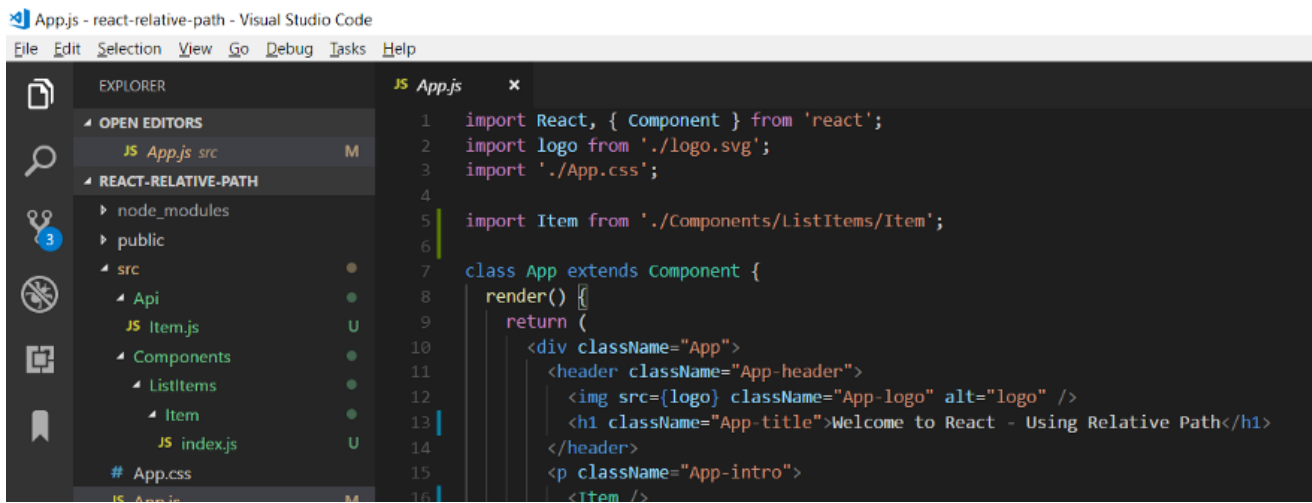




And a folder called **Components** with subfolders **ListItems** and **Item** as we can see below:



On **App.js** we just import and use the component **Item**:



```
registerServiceWorker.js 22
.gitignore 23 export default App;
App.js 24
```

The result should be like this:

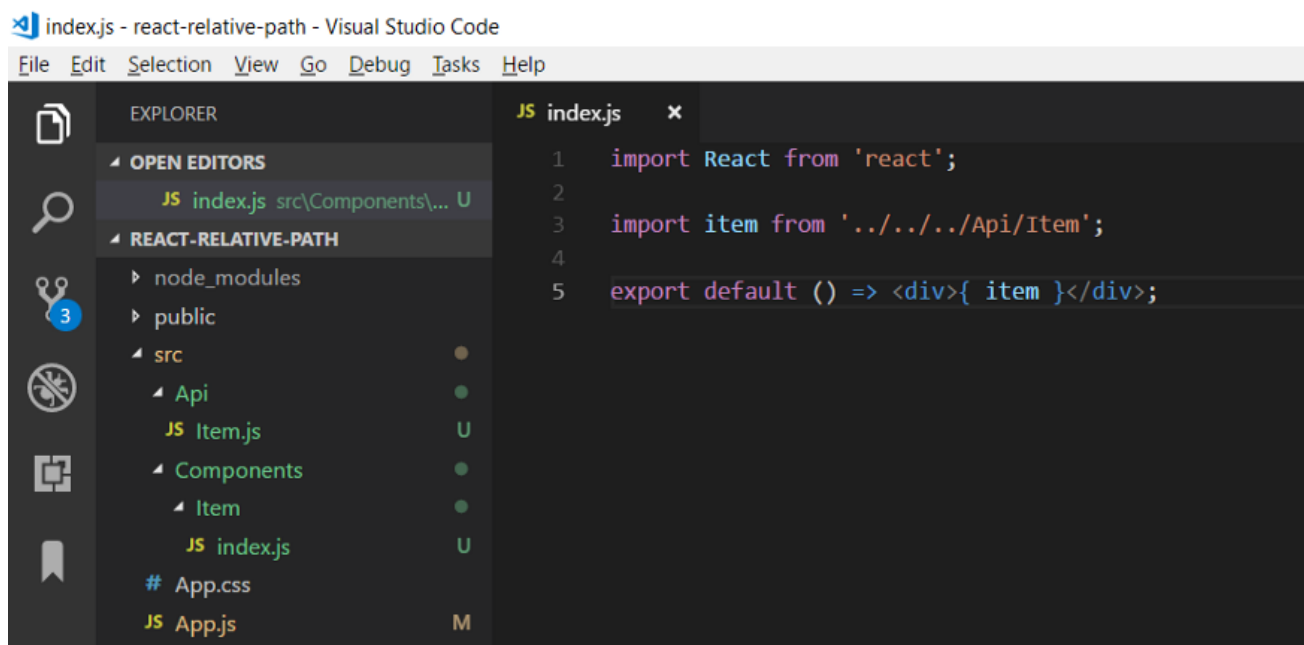


item 1

Alright, everything is working well, but if you decide to change the folder structure? Remember the path inside Item component:

```
import item from '../../../Api/Item';
```

If you decide to remove the folder **ListItems** and let only the folder **Item**:



## Failed to compile

```
./src/Components/ListItem/Item/index.js  
Module build failed: Error: ENOENT: no such file or directory, open 'C:\Temp\React\react-relative-path\src\Components\ListItem\Item\index.js'
```

This error occurred during the build time and cannot be dismissed.

We have to change all paths inside all components inside **Item** folder, of course in this case is easy, but imagine you have many components, I could be hard to maintain.

. . .

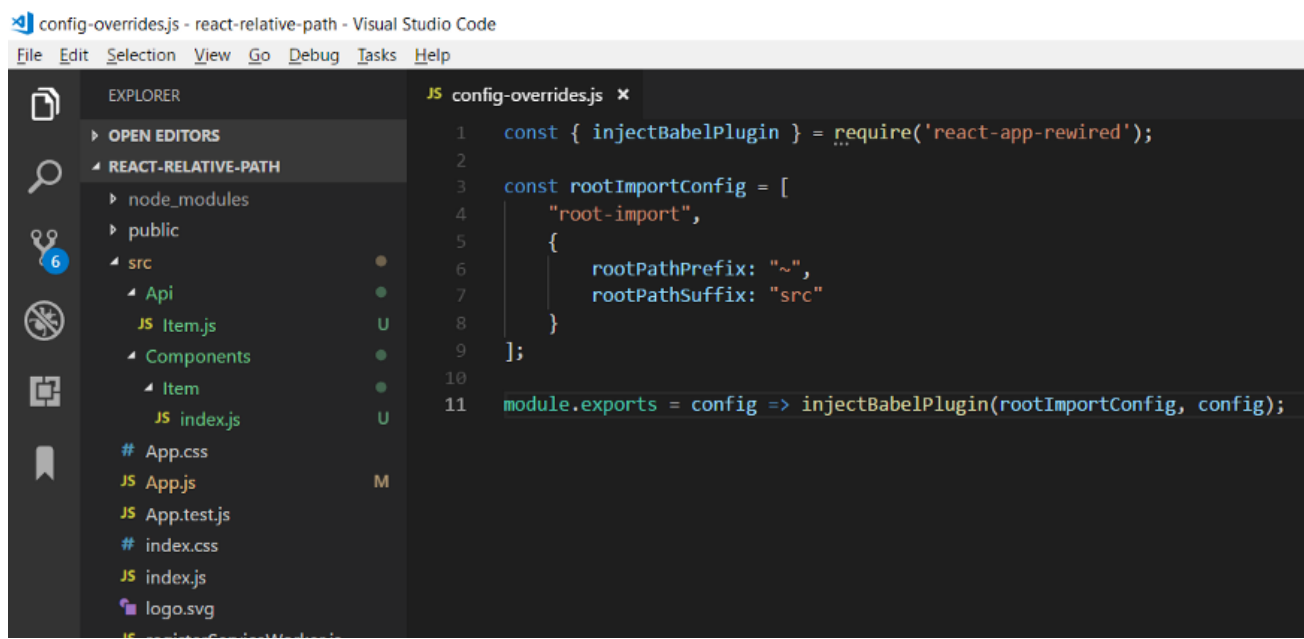
In order to deal with this situation, we can use some third-part components. This first we need to install is the **react-app-rewired**. This component allow us to change webpack configuration without using 'eject' and without creating a fork of the react-scripts:

```
npm i --dev react-app-rewired
```

The second component the **babel-plugin-root-import**, that allow us to import components using root based paths.

```
npm i --dev babel-plugin-root-import
```

Alright, now we need create a file on root folder called **config-overrides.js**:





package.json M  
README.md

In this file we are going to configure basically two things: the root prefix and root suffix. The **rootPathPrefix** is the character we would like to use to refer the root folder and the **rootPathSuffix** is the name of the root folder.

Next step is change the **package.json** file to use the **react-app-rewired** (except the **eject script**):

package.json - react-relative-path - Visual Studio Code

File Edit Selection View Go Debug Tasks Help

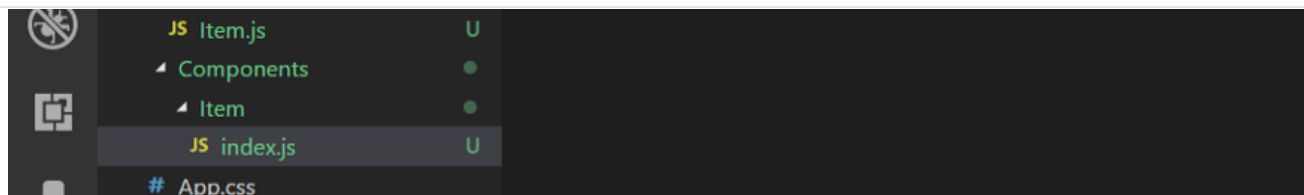
```
{
  "name": "react-relative-path",
  "version": "0.1.0",
  "private": true,
  "dependencies": {
    "babel-plugin-root-import": "^6.1.0",
    "react": "^16.4.2",
    "react-app-rewired": "^1.5.2",
    "react-dom": "^16.4.2",
    "react-scripts": "1.1.4"
  },
  "scripts": {
    "start": "react-app-rewired start",
    "build": "react-app-rewired build",
    "test": "react-app-rewired test --env=jsdom",
    "eject": "react-scripts eject"
  }
}
```

Finally let's change the **Item** component to use the new prefix "~":

index.js - react-relative-path - Visual Studio Code

File Edit Selection View Go Debug Tasks Help

```
import React from 'react';
import item from '~/Api/Item';
```



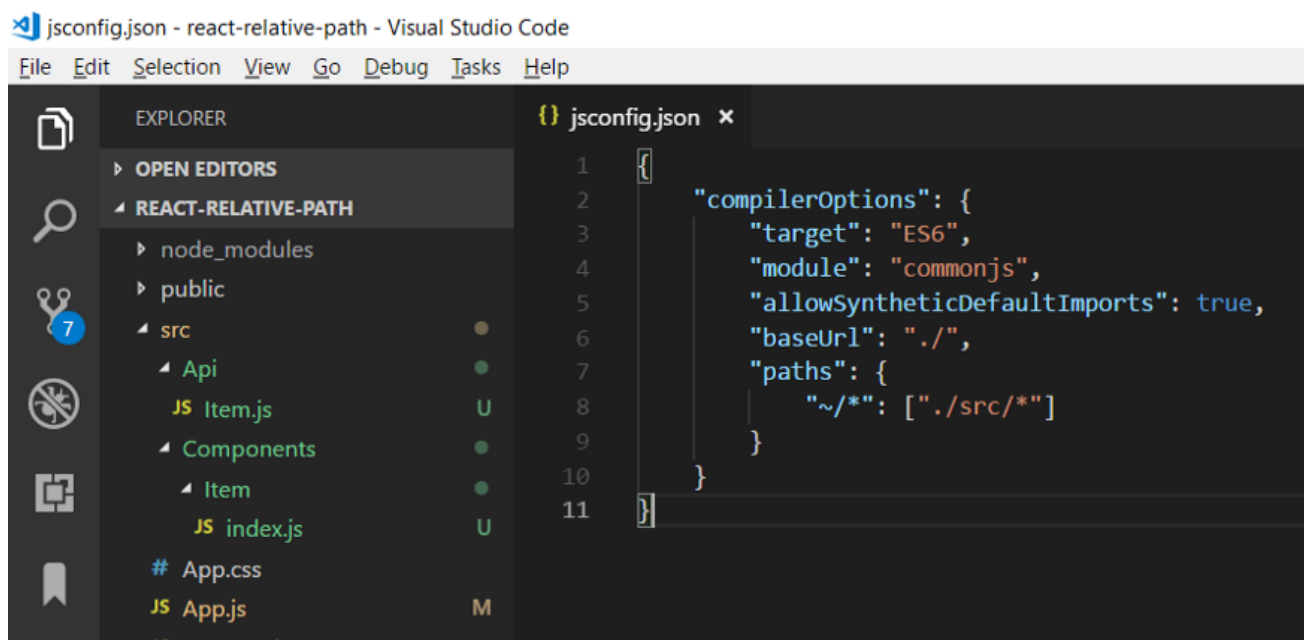
And now it's working again:

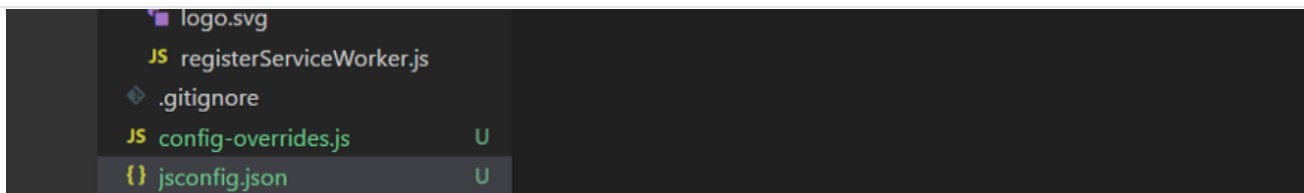


item 1

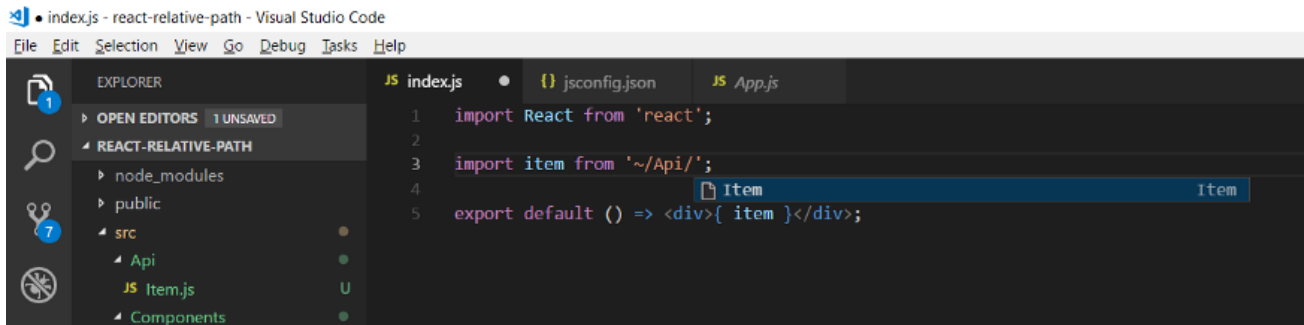
. . .

If you are using VSCode, I have one more tip. After these changes we've made, you can see we lost the intellisense for paths. So, after you type `~/...` the VSCode will not complete or suggest the available path. To fix this, we need to add a file called **jsconfig.json** and add some configs:





The most important part is the “paths” section. In this section we instruct the VSCode to map the path “~/” to “./src” and after that the intellisense will work again:



You can clone this project here: <https://github.com/lblima/react-relative-path>

That’s all folks,

Thanks for reading!

### timarney/react-app-rewired

react-app-rewired - Override create-react-app webpack configs without ejecting

[github.com](https://github.com)



### entwicklerstube/babel-plugin-root-import

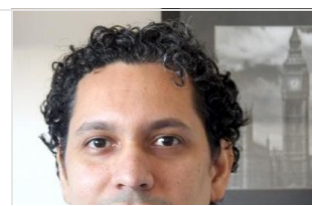
babel-plugin-root-import - Add the opportunity to import modules by the root path

[github.com](https://github.com)



### lblima/react-relative-path

GitHub is where people build software. More than 28 million people use GitHub to discover, fork, and contribute to over...





Open in app



JavaScript

React

Reactjs

Babel

Babeljs



[About](#) [Help](#) [Legal](#)

Get the Medium app

