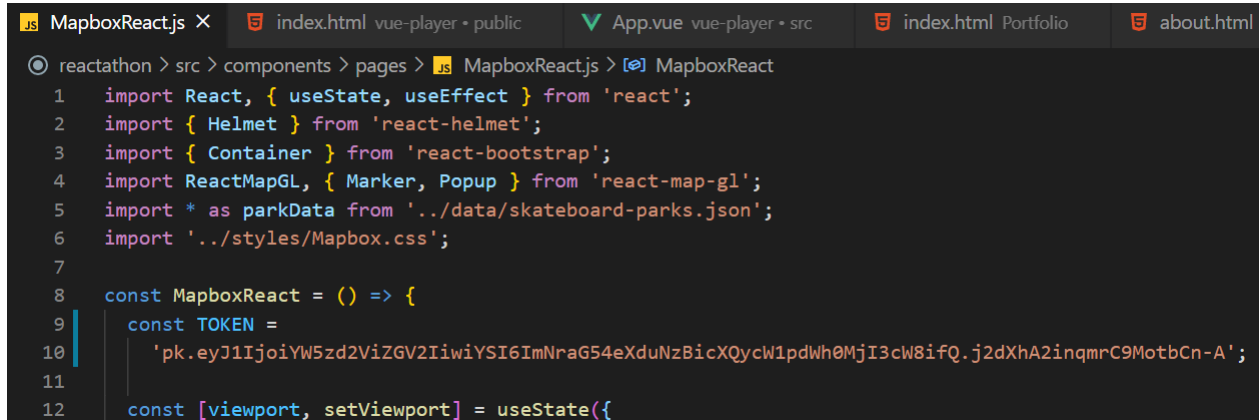


USING MAPBOX API TOKENS (TROUBLESHOOTING)

Setting up a .env.local File

(to skip to troubleshooting, scroll down to the **Troubleshooting** section).

If you use an API token in your project, one way to go about it is to set it in a variable in your React component, and then use it in the same file. In this case, it's set the variable in lines 9-10.



```
1 import React, { useState, useEffect } from 'react';
2 import { Helmet } from 'react-helmet';
3 import { Container } from 'react-bootstrap';
4 import ReactMapGL, { Marker, Popup } from 'react-map-gl';
5 import * as parkData from '../data/skateboard-parks.json';
6 import '../styles/Mapbox.css';
7
8 const MapboxReact = () => {
9   const TOKEN =
10     'pk.eyJ1IjoiYW5zd2ViZGV2IiwiaYSI6ImNraG54eXduZBicXQycW1pdWh0MjI3cW8ifQ.j2dXhA2inqmrc9MotbCn-A';
11
12   const [viewport, setViewport] = useState({
```

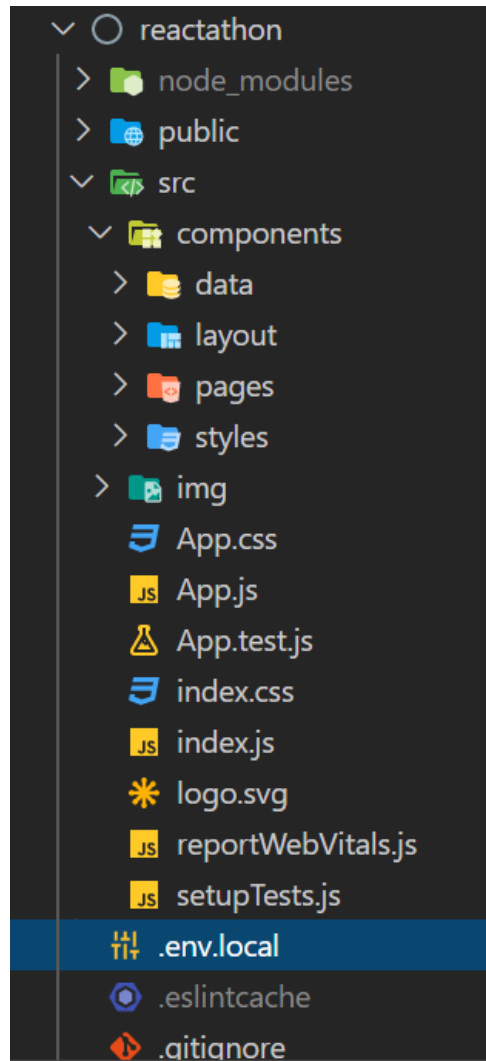
And then, it is used down below (line 167).



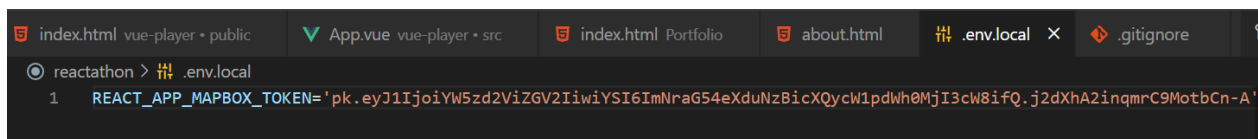
```
164 <div style={{ marginBottom: '40px' }}>
165   <ReactMapGL
166     {...viewport}
167     mapboxApiAccessToken={TOKEN}
168     mapStyle='mapbox://styles/answebdev/ckhp3rta60d2l19piwffs1aj5'
169     onViewportChange={(viewport) => {
170       setViewport(viewport);
171     }}
172   >
```

However, this is not ideal. If you are using your own token, you may want to keep it secret and private, so that it does not show publicly in your code, or show up in your public GitHub repo.

In order to keep your token secret, create a file called **.env.local** in the root.



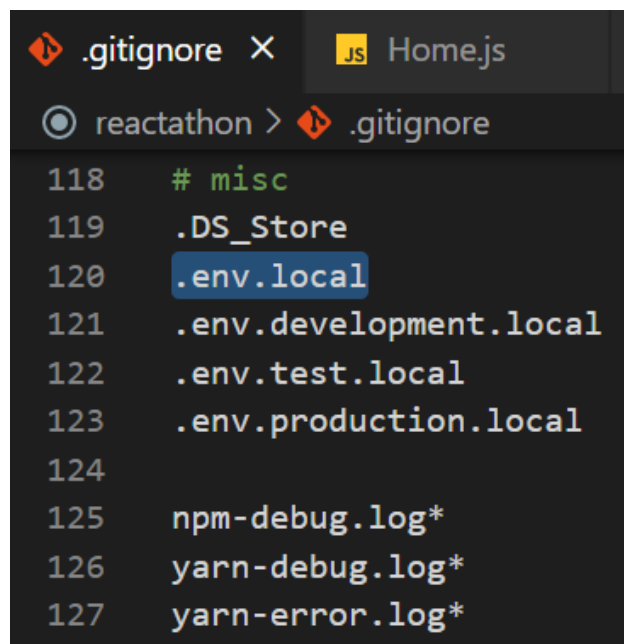
Then, put your token in this file.



Next, in your code (i.e. your React component), instead creating a variable and setting the token to that, simply reference it in your code using **process.env** and the name of the key (in this case, **REACT_APP_MAPBOX_TOKEN**) (line 167). And that's it. Notice that your code no longer contains the actual token, so it will not be publicly visible.

```
164 <div style={{ marginBottom: '40px' }}>
165   <ReactMapGL
166     {...viewport}
167     mapboxApiAccessToken={process.env.REACT_APP_MAPBOX_TOKEN}
168     mapStyle='mapbox://styles/answebdev/ckhp3rta60d2l19piwffs1aj5'
169     onViewportChange={(viewport) => {
170       setViewport(viewport);
171     }}
172   >
```


Finally, add your **.env.local** file in your **.gitignore** file, if it is not already in there.



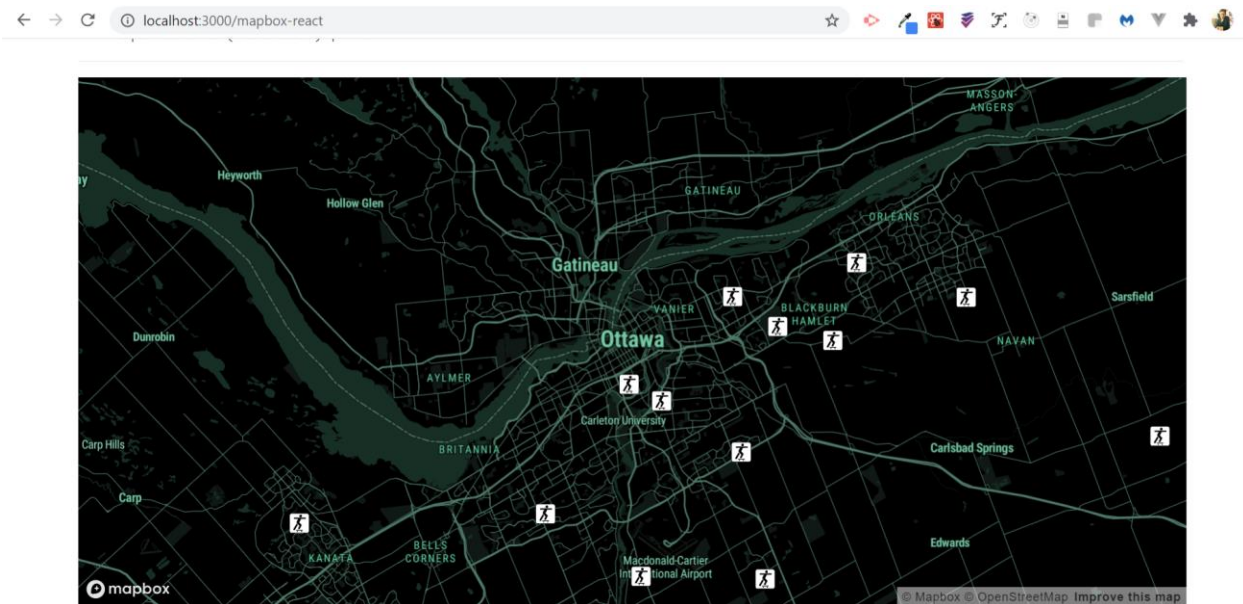
The screenshot shows a code editor with a dark theme. The top bar has a file icon, the name '.gitignore', a close button 'X', and a tab for 'Home.js'. Below the top bar, the breadcrumb 'reactathon > .gitignore' is visible. The main editor area shows the contents of the '.gitignore' file. Line 118 is '# misc'. Line 119 is '.DS_Store'. Line 120 is '.env.local', which is highlighted with a blue selection box. Line 121 is '.env.development.local'. Line 122 is '.env.test.local'. Line 123 is '.env.production.local'. Line 124 is empty. Line 125 is 'npm-debug.log*'. Line 126 is 'yarn-debug.log*'. Line 127 is 'yarn-error.log*'. Line numbers 118 through 127 are visible on the left side of the editor.

```
118 # misc
119 .DS_Store
120 .env.local
121 .env.development.local
122 .env.test.local
123 .env.production.local
124
125 npm-debug.log*
126 yarn-debug.log*
127 yarn-error.log*
```

That's it. And now, since your token in your **.env.local** file is in your **.gitignore** file, it will not show up in your public repo, which is what you want. And now, it is not public, so nobody can see your token.

 answebdev	added .env.local file	d77eb14 26 minutes ago	🕒 16 commits
📁 public	added Open Graph Tags		17 days ago
📁 src	added .env.local file		26 minutes ago
📄 .gitignore	initial commit		18 days ago
📄 README.md	initial commit		18 days ago
📄 README.old.md	initial commit		18 days ago
📄 package-lock.json	added react-spring		18 days ago
📄 package.json	added react-spring		18 days ago

And you can see that it still works. Since we are using the Mapbox token here, we can see our map in the browser.

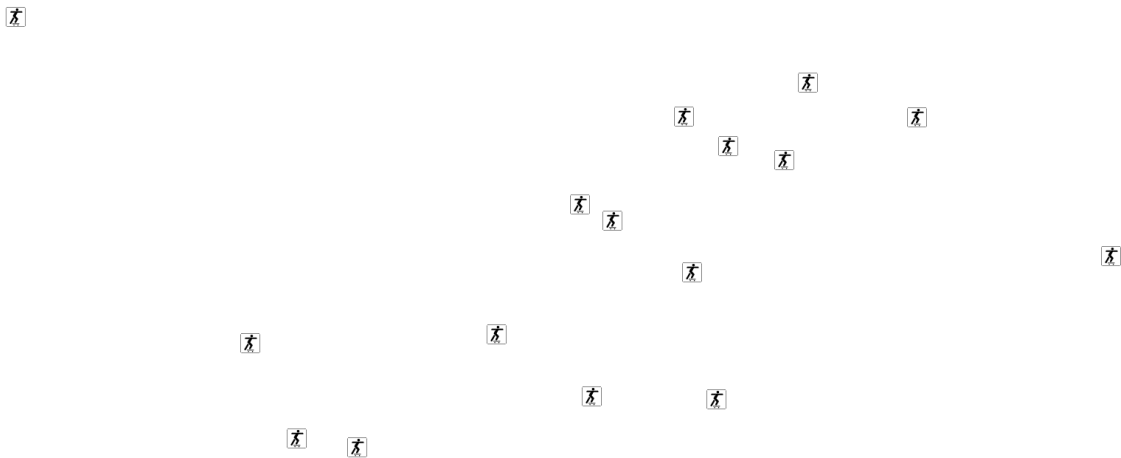


Troubleshooting

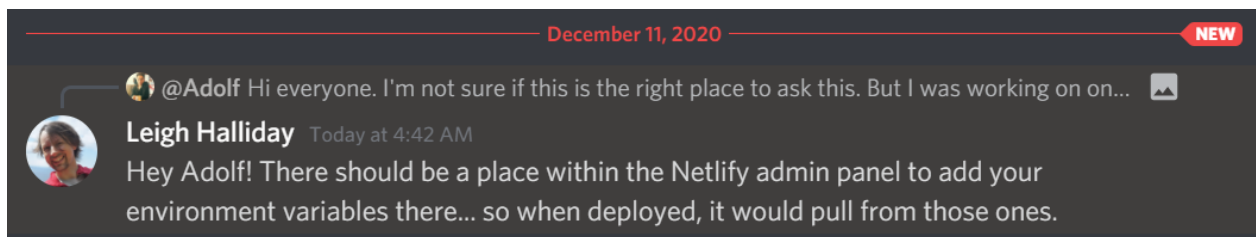
So, everything works as it should. But note, that everything is working *locally*. Let's say that you want to host and deploy this on Netlify, for example. Now after pushing up your changes to GitHub and publishing your site on Netlify, you can no longer see your map in the browser. Instead, you get this:


NO_TOKEN_WARNING




For information on setting up your basemap, read
[Note on Map Tokens](#)



Since Netlify is being used to host the site, the environment variables must be added within the Netlify admin panel.



britvaica's team > react-a-thon






Site overview **Deploys** Plugins Functions Identity Forms Large Media Split Testing Analytics Site settings

Deploys for react-a-thon

- <https://react-a-thon.netlify.app>

github.com/answebdev/reactathon, published main@HEAD.

Auto publishing is on. Deploys from main are published automatically.

 Deploy settings  Notifications  Stop auto publishing

General

Build & deploy

Continuous Deployment

Environment

Post processing

Deploy notifications

Domain management

Analytics

Functions

Identity

Forms

Large Media

Environment

Control the environment your site builds in and/or gets deployed to.

Environment variables

Set environment variables for your build script and add-ons.

Key

Value

REACT_APP_MAPBOX_TOKEN

pk.eyJ1IjoiYW5zd2ViZGV2liwiYSI6ImNraG54eXdu

New variable

[Learn more about environment variables in the docs](#)

Save

Cancel

index.html vue-player + public App.vue vue-player + src index.html Portfolio about.html .env.local .gitignore

```
reactathon > .env.local
1  REACT_APP_MAPBOX_TOKEN='pk.eyJ1IjoieWV5Z2ViZGV2IiwiaYSi6ImNraG54eXduZnBicXQycW1pdWh0MjI3cW81fQ.j2dXhA2inqmrC9MotbCn-A'
```

That's it. Once you do that, deploy the site again. And it should now work. And now when you go to the browser, you should get your map (note URL – site is live).

