

coin-teer project + error

태그	
날짜	@2023년 4월 24일

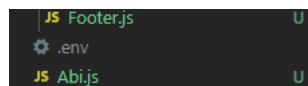
지금까지 git을 사용하지 않았었는데 후회가 된다.

git을 사용해야한다.

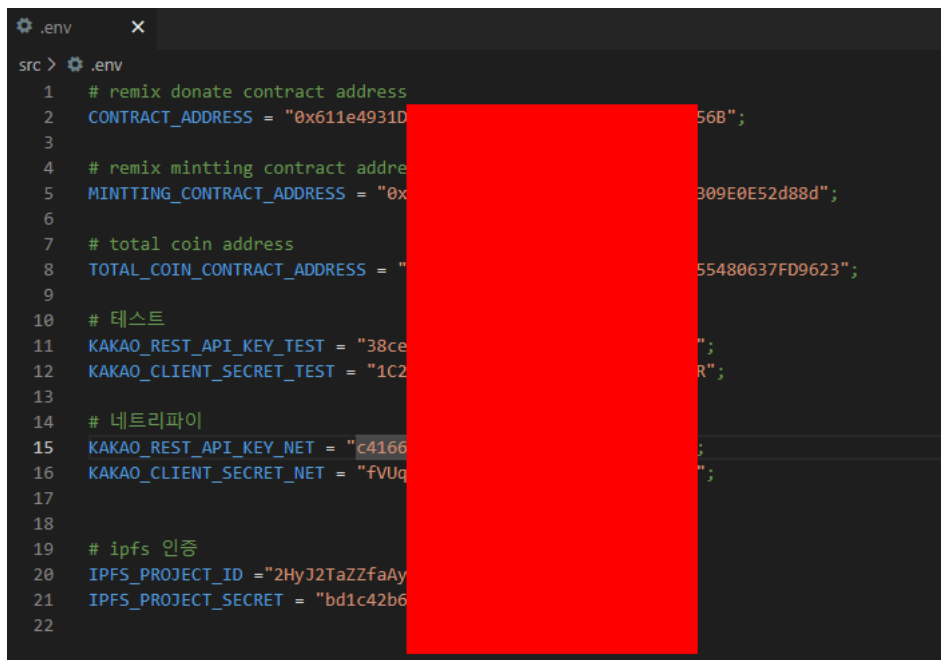
git을 사용해야하니 중요한 컨트랙트 주소나 중요 정보를 따로 빼야한다는걸 알았다.

dotenv라는 것을 알게되어 설치를 하고 중요 정보를 .env파일로 옮겼다.

- npm install dotenv
- 루트 경로에 .env 파일 만들기



- 중요 key값 추가



- .gitignore 에 .env 추가하기

```

.gitignore
1 # See https://help.github.com/articles/ignoring-files
2
3 # dependencies
4 /node_modules
5 /.pnp
6 .pnp.js
7
8 # testing
9 /coverage
10
11 # production
12 /build
13
14 # misc
15 .DS_Store
16 .env.local
17 .env.development.local
18 .env.test.local
19 .env.production.local
20 .env
21
22 npm-debug.log*
23 yarn-debug.log*
24 yarn-error.log*
25

```

dotenv로 중요 정보를 .env로 옮긴 후 npm run build 에러가 생겼다.

error

```

C:\Users\User\OneDrive\바탕 화면\minho-cording\4th-project-230328>npm run build

> 4th-project@0.1.0 build
> react-scripts build

Creating an optimized production build...
Failed to compile.

Module not found: Error: Can't resolve 'path' in 'C:\Users\User\OneDrive\바탕 화면\minho-cording\4th-project-230328\node_modules\dotenv\lib\polyfills'
BREAKING CHANGE: webpack < 5 used to include polyfills for node.js core modules by default.
This is no longer the case. Verify if you need this module and configure a polyfill for it.

If you want to include a polyfill, you need to:
  - add a fallback 'resolve.fallback: { "path": require.resolve("path-browserify") }'
  - install 'path-browserify'
If you don't want to include a polyfill, you can use an empty module like this:
  resolve.fallback: { "path": false

```

추가적으로 아래 에러가 생겼다.

긴 에러지만 아래 에러들은 생략했다.

```

Module not found: Error: Can't resolve 'fs' in 'C:\Users\User\OneDrive\바탕 화면\minho-cording\4th-project-230328\node_modules\dotenv\lib\polyfills'
WARNING in ./node_modules/web3/dist/web3.min.js
Module Warning (from ./node_modules/source-map-loader/dist/cjs.js):
Failed to parse source map: 'webpack:///./src.ts/coders/abstract-coder.ts' URL is not support

```

- 구글링 해보니 로컬에 있는 node_modules에서 해당하는 파일들을 찾을 수 없어서 뜨는 경고문들이다.
- webpack과 CRA5의 문제라고 하여 .env 파일에 `GENERATE_SOURCEMAP=false`를 추가했다.
- 추가 하고 나니 창은 열리지만 다른 에러가 생겼다.

CRA, 즉 create-react-app 으로 만든 React 프로젝트에는 이미 dotenv 패키지가 내장되어 있다고 한다.

따라서 별도의 패키지 추가나 Webpack에 대한 설정 없이, .env 파일을 생성해서 변수를 선언하는 것만으로도 환경변수를 사용할 수 있다고 한다.

CRA가 내부적으로 REACT_APP_으로 시작하는 환경변수만 읽어 들이도록 설정을 해주었기 때문에 변수명은 반드시 REACT_APP_으로 시작해야하니 주의하자.

```
.env
src > .env
1  GENERATE_SOURCEMAP=false
2
3  # remix donate contract address
4  REACT_APP_CONTRACT_ADDRESS = "0x611e4931D...DFcA56B";
5
6  # remix mintting contract address
7  REACT_APP_MINTTING_CONTRACT_ADDRESS = "0x...5309309E0E52d88d
8
9  # total coin address
10 REACT_APP_TOTAL_COIN_CONTRACT_ADDRESS = "
11
12 # 테스트
13 REACT_APP_KAKAO_REST_API_KEY_TEST = "38ce...57a9";
14 REACT_APP_KAKAO_CLIENT_SECRET_TEST = "1C2...vxRfR";
15
16 # 네트리파이
17 REACT_APP_KAKAO_REST_API_KEY_NET = "c4166...d31";
18 REACT_APP_KAKAO_CLIENT_SECRET_NET = "fVUq...DLsL";
19
20
21 # ipfs 인증
22 REACT_APP_IPFS_PROJECT_ID ="2HyJ2TaZZfaAy
23 REACT_APP_IPFS_PROJECT_SECRET = "bd1c42b6
24
```

모두 변경했다. 하지만 에러는 여전히 남아있었다.

구글링을 다시 해보니

CRA 로 생성한 프로젝트는 기본적으로 dotenv가 내장되어 있기 때문에 무지하게 dotenv를 import 했기 때문에 생긴 에러 였다.

다시 npm run build 를 하니 에러가 없어졌다.

네트리파이에 다시 업로드 후 확인.

네트리파이에 업로드 후 확인 결과 홈 렌더링은 잘 되었다.

하지만 카카오 로그인에 에러가 발생했다.

console.log(REST_API_KEY) 로 확인해보니 .env에서 값을 가져오지 못했다.

process.env는 실행시 로드 된다고 한다.

.env 설정을 바꾸게 되면 React 프로젝트를 다시 구동하여야한다.

다시 찾아보니 .env 파일을 root 경로가 아닌 /src 아래에 생성해서 발생한 에러였다.

```
⚠ Mixed Content: The page at 'https://coin-tee /Check Coin:1
r.netlify.app/Check Coin' was loaded over HTTPS, but
requested an insecure element 'http://k.kakaocdn.net/dn/bE0
z5t/btrZb0FXNnr/v6do8Xb0YzfJVI36aLeGK/img_640x640.jpg'.
This request was automatically upgraded to HTTPS, For more
information see https://blog.chromium.org/2019/10/no-more-m
ixed-messages-about-https.html
```

다시 위와 같은 에러가 발생했다.

https 사이트에서 ajax를 사용해서 비동기로 http 사이트에 request를 요청해서 문제가 발생 했다고한다. 암호화된 HTTPS 기반의 사이트에서 암호화되지 않은 HTTP 사이트에 요청을 보낼 때 Mixed content 에러가 발생한다고 한다.

Mixed content 에러 해결

- html파일 head 에 추가

```
<meta http-equiv="Content-Security-Policy" content="upgrade-insecure-requests">
```