day85-sol-tokenproject2

태그	
날짜	@2023년 1월 27일

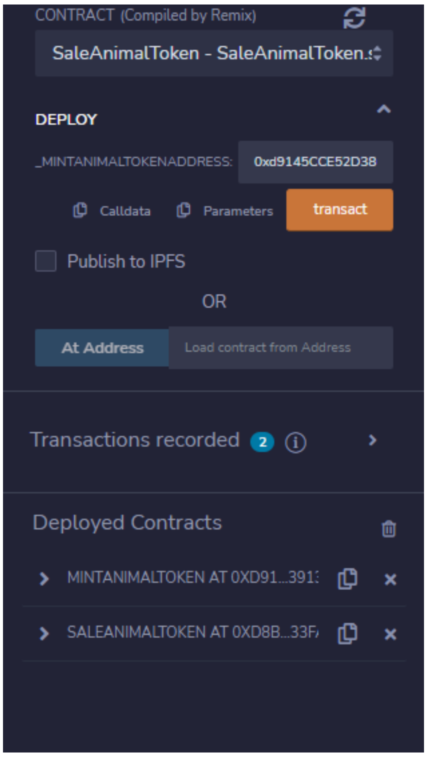
스마트 컨트렉트

NFT 발행

판매 구매

SaleAnimalToken.sol

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.7;


import "MintAnimalToken.sol";

contract SaleAnimalToken{
    MintAnimalToken public mintAnimalTokenAddress;

    constructor(address _mintAnimalTokenAddress){
        mintAnimalTokenAddress = MintAnimalToken(_mintAnimalTokenAddress);
    }

    mapping(uint256=>uint256) public animalTokenPrices;

    uint256[] public onSaleAnimalTokenArray;

    function setForSaleAnimalToken(uint256 _animalTokenId, uint256 _price) public{
        //토큰 주인
        address animalTokenOwner = mintAnimalTokenAddress.ownerOf(_animalTokenId);

        //토큰 주인이 현재 메타마스크 sign 하는 사람하고 같으면 통과
        require(animalTokenOwner == msg.sender, "Caller is not animal token owner.");
        require(_price > 0, "Price is zero or lower.");
        require(animalTokenPrices[_animalTokenId] == 0, "This animal token is already on sale.");
        require(mintAnimalTokenAddress.isApprovedForAll(animalTokenOwner, address(this)), "Animal token owner did not approve token.")

        animalTokenPrices[_animalTokenId] = _price;

        onSaleAnimalTokenArray.push(_animalTokenId);
    }

    function puschaseAnimalToken(uint256 _animalTokenId) public payable{
        uint256 price = animalTokenPrices[_animalTokenId];
        address animalTokenOwner = mintAnimalTokenAddress.ownerOf(_animalTokenId);

        require(price > 0, "Animal token not sale.");
        require(price <= msg.value , "Caller sent lower than price.");
        require(animalTokenOwner != msg.sender , "Caller is animal token owner.");

        payable(animalTokenOwner).transfer(msg.value);
        mintAnimalTokenAddress.safeTransferFrom(animalTokenOwner, msg.sender, _animalTokenId);

        animalTokenPrices[_animalTokenId] = 0;

        for(uint256 i = 0; i < onSaleAnimalTokenArray.length; i++){
            if(animalTokenPrices[onSaleAnimalTokenArray[i]] == 0) {
                onSaleAnimalTokenArray[i] = onSaleAnimalTokenArray[onSaleAnimalTokenArray.length - 1];
                onSaleAnimalTokenArray.pop();
            }
        }

    }
    function getOnSaleAnimalTokenArrayLength() view public returns (uint256) {
        return onSaleAnimalTokenArray.length;
    }
}
```



CONTRACT (Compiled by Remix)

SaleAnimalToken - SaleAnimalToken.s

DEPLOY

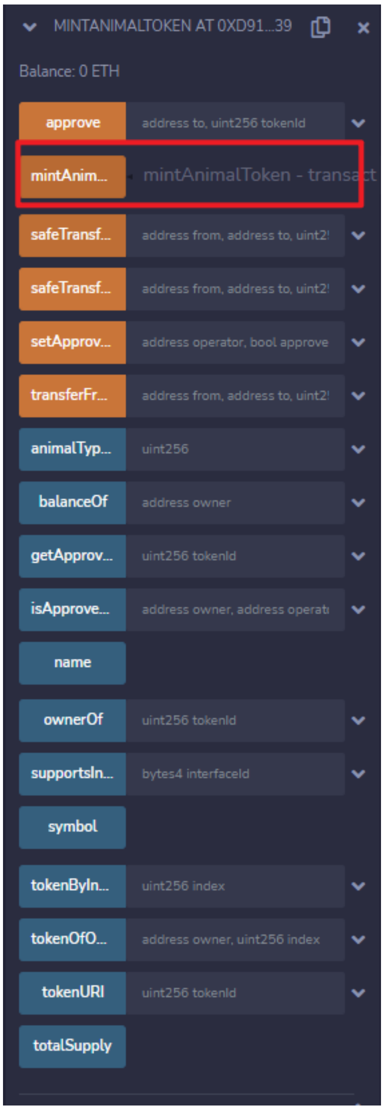
_MINTANIMALTOKENADDRESS: 0xd9145CCE52D38

Deployed Contracts

MINTANIMALTOKEN AT 0XD91...391:

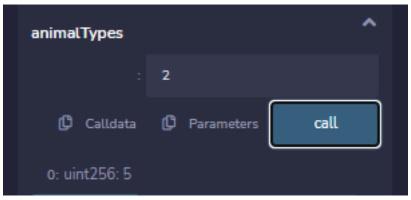
SALEANIMALTOKEN AT 0XD8B...33F/

토큰 발행


MINTANIMALTOKEN AT 0XD91...39

Balance: 0 ETH

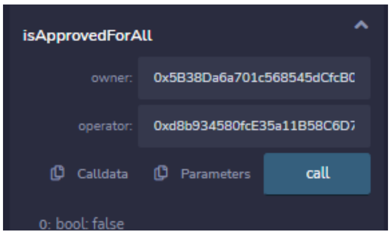
발행확인


animalTypes

: 2

0: uint256: 5

sale 컨트렉트 복사


isApprovedForAll

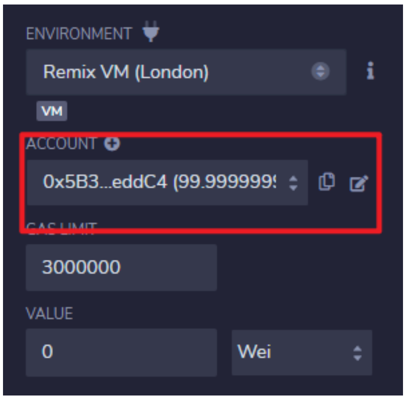
owner: 0x5B38Da6a701c568545dCfcB0

operator: 0xd8b934580fcE35a11B58C6D7

0: bool: false

승인 확인

owner :


ENVIRONMENT

Remix VM (London)

VM

ACCOUNT

0x5B3...eddC4 (99.999999!

GAS LIMIT

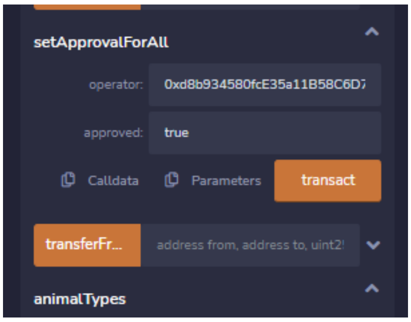
3000000

VALUE

0

Wei

operate: sale 컨트렉트


setApprovalForAll

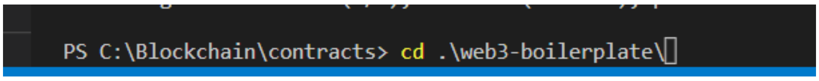
operator: 0xd8b934580fcE35a11B58C6D7

approved: true

sale 컨트렉트를 true로 승인

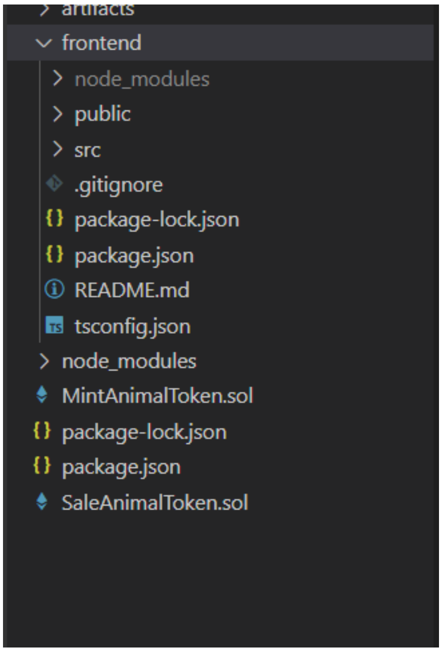
- git clone https://github.com/h662/web3-boilerplate

명령어로 설치


```
PS C:\Blockchain\contracts> cd .\web3-boilerplate\
```

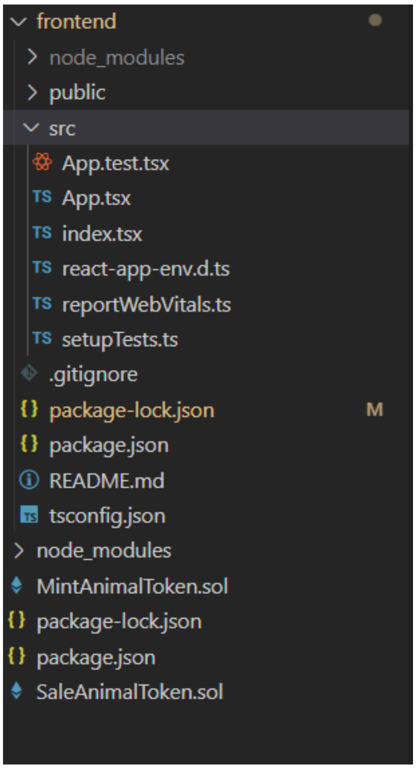
폴더명을 frontend로 변경

설치 완료



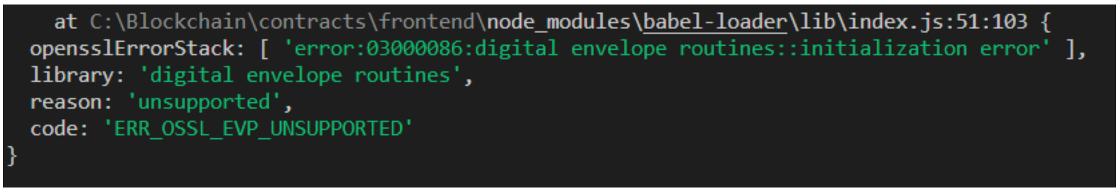
```
> artifacts
v frontend
  > node_modules
  > public
  > src
    .gitignore
    package-lock.json
    package.json
    README.md
    tsconfig.json
> node_modules
  MintAnimalToken.sol
  package-lock.json
  package.json
  SaleAnimalToken.sol
```

- cd frontend
- npm install



```
v frontend                     ●
  > node_modules
  > public
  v src
      App.test.tsx
      App.tsx
      index.tsx
      react-app-env.d.ts
      reportWebVitals.ts
      setupTests.ts
    .gitignore
    package-lock.json          M
    package.json
    README.md
    tsconfig.json
> node_modules
  MintAnimalToken.sol
  package-lock.json
  package.json
  SaleAnimalToken.sol
```

설치 완료



```
    at C:\Blockchain\contracts\frontend\node_modules\babel-loader\lib\index.js:51:103 {
  opensslErrorStack: [ 'error:03000086:digital envelope routines::initialization error' ],
  library: 'digital envelope routines',
  reason: 'unsupported',
  code: 'ERR_OSSL_EVP_UNSUPPORTED'
}
```

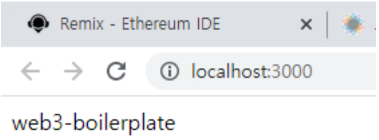
노드 버전 에러

16버전 설치 이후

node -v 로 버전 확인

npm install

npm start



web3-boilerplate

구글 chakra ui 검색

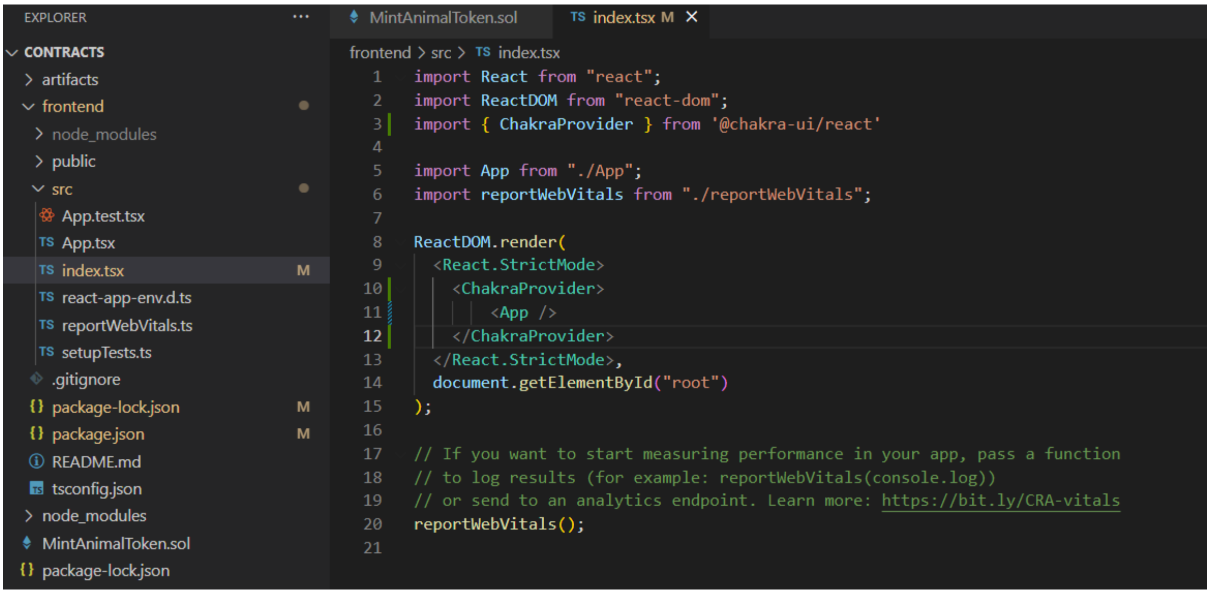
install 부분 카피

- npm i @chakra-ui/react @emotion/react@^11 @emotion/styled@^11 framer-motion@^5 —force

차크라유아이 설치

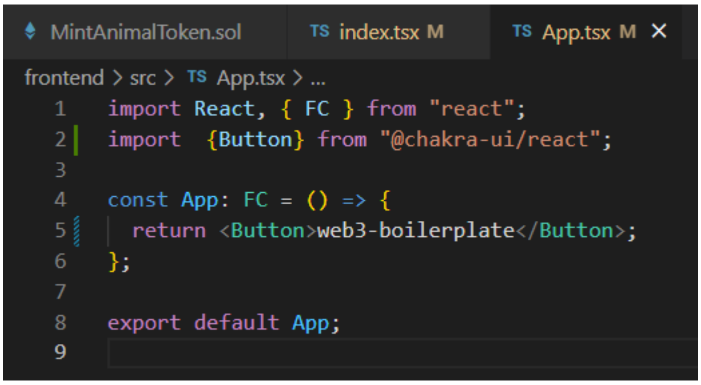
src/index.tsx

```
import React from "react";
import ReactDOM from "react-dom";
import { ChakraProvider } from '@chakra-ui/react'

import App from "./App";
import reportWebVitals from "./reportWebVitals";

ReactDOM.render(
  <React.StrictMode>
    <ChakraProvider>
        <App />
    </ChakraProvider>
  </React.StrictMode>,
  document.getElementById("root")
);

// If you want to start measuring performance in your app, pass a function
// to log results (for example: reportWebVitals(console.log))
// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
reportWebVitals();
```



```
import React from "react";
import ReactDOM from "react-dom";
import { ChakraProvider } from '@chakra-ui/react'

import App from "./App";
import reportWebVitals from "./reportWebVitals";

ReactDOM.render(
  <React.StrictMode>
    <ChakraProvider>
        <App />
    </ChakraProvider>
  </React.StrictMode>,
  document.getElementById("root")
);

// If you want to start measuring performance in your app, pass a function
// to log results (for example: reportWebVitals(console.log))
// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
reportWebVitals();
```

App.tsx



```
import React, { FC } from "react";
import  {Button} from "@chakra-ui/react";

const App: FC = () => {
  return <Button>web3-boilerplate</Button>;
};

export default App;
```