

day77-sol-Dapp

≡ 태그	
📅 날짜	@2023년 1월 17일

리액트 + 솔리디티 블록체인

명언 dapp

1.

```
// SPDX-License-Identifier: GPL-3.0
pragma solidity >=0.8.0 <0.9.0;

contract SayContract{
    event AddSay(address recipient, uint sayId);
    event DeleteSay(uint sayId, bool isDeleted);

    struct Say{
        uint id;
        address username;
        string author;
        string sayText;
        bool isDeleted;
    }

    Say[] private says;

    mapping(uint256 => address) sayToOwner;

    function addSay(string memory author, string memory sayText, bool isDeleted) external
    {
        uint sayId = says.length;
        says.push(Say(sayId, msg.sender, author, sayText, isDeleted));
        sayToOwner[sayId] = msg.sender;

        emit AddSay(msg.sender, sayId);
    }

    function getMySays() external view returns(Say[] memory){
        Say[] memory temporary = new Say[](says.length);
        uint counter = 0;
        for(uint i = 0; i < says.length; i++){
            if(says[i].isDeleted == false){
                temporary[counter] = says[i];
                counter++;
            }
        }
        Say[] memory result = new Say[](counter);
        for(uint i = 0; i < counter; i++){
            result[i] = temporary[i];
        }
        return result;
    }

    function deleteSay(uint sayId, bool isDeleted) external {
        if(sayToOwner[sayId] == msg.sender) {
            says[sayId].isDeleted = isDeleted;
            emit DeleteSay(sayId, isDeleted);
        }
    }
}
```

컨트랙트 주소

0x588b0b5ef24bB5A87bc695B12eA8A4CbF11acEF

리액트 폴더 생성

- npx create-react-app blocksay
- cd blocksay

- code .

web3설치

- npm install web3

라우트 설치

- npm install react-router-dom

index.js

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';
import { BrowserRouter } from 'react-router-dom';

root.render(
  <BrowserRouter>
    <App />
  </BrowserRouter>,
  document.getElementById('root')
);
```

app.js

```
import React, {Component} from "react";
import {Route, Routes} from "react-router-dom";

class App extends Component {

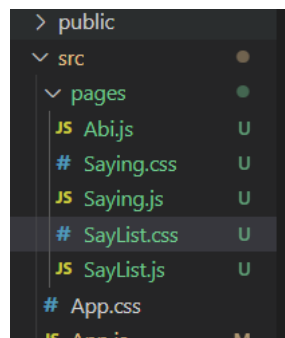
  render(){

    return (
      <Routes>

      </Routes>
    );
  }
}

export default App;
```

페이지 구성



remix 에서 Abi 가져오기

```
const Abi = [
  {
    "inputs": [
      {
        "internalType": "string",
        "name": "author",
        "type": "string"
      },
      {
        "internalType": "string",
        "name": "sayText",
        "type": "string"
      },
      {
        "internalType": "bool",
        "name": "isDeleted",
        "type": "bool"
      }
    ],
    "name": "addSay",
    "outputs": [],
    "stateMutability": "nonpayable",
    "type": "function"
  },
  {
    "anonymous": false,
    "inputs": [
      {
        "indexed": false,
        "internalType": "address",
        "name": "recipient",
        "type": "address"
      },
      {
        "indexed": false,
        "internalType": "uint256",
        "name": "sayId",
        "type": "uint256"
      }
    ],
    "name": "AddSay",
    "type": "event"
  },
  {
    "inputs": [
      {
        "internalType": "uint256",
        "name": "sayId",
        "type": "uint256"
      },
      {
        "internalType": "bool",
        "name": "isDeleted",
        "type": "bool"
      }
    ],
    "name": "deleteSay",
    "outputs": [],
    "stateMutability": "nonpayable",
    "type": "function"
  },
  {
    "anonymous": false,
    "inputs": [
      {
        "indexed": false,
        "internalType": "uint256",
        "name": "sayId",
        "type": "uint256"
      },
      {
        "indexed": false,
        "internalType": "bool",
        "name": "isDeleted",
        "type": "bool"
      }
    ],
    "name": "DeleteSay",
    "type": "event"
  },
  {
    "inputs": [],
    "name": "getMySays",
```

```

"outputs": [
  {
    "components": [
      {
        "internalType": "uint256",
        "name": "id",
        "type": "uint256"
      },
      {
        "internalType": "address",
        "name": "username",
        "type": "address"
      },
      {
        "internalType": "string",
        "name": "author",
        "type": "string"
      },
      {
        "internalType": "string",
        "name": "sayText",
        "type": "string"
      },
      {
        "internalType": "bool",
        "name": "isDeleted",
        "type": "bool"
      }
    ],
    "internalType": "struct SayContract.Say[]",
    "name": "",
    "type": "tuple[]"
  }
],
"stateMutability": "view",
"type": "function"
}
]

export default Abi;

```

Saying.js

```

import Web3 from "web3";
import React, {useRef, useState, useEffect} from "react";
import { Link } from "react-router-dom";
import "./Saying.css";
import Abi from "../Abi";

const Saying = () => {
  return(
    <div>

    </div>
  );
}

export default Saying;

```

SayList.js

```

import Web3 from "web3";
import React, {useRef, useState, useEffect} from "react";
import { Link } from "react-router-dom";
import "./SayList.css";
import Abi from "../Abi";

const SayList = () => {
  return(
    <div>

    </div>
  );
}

```

```

}

export default SayList;

```

app.js

```

import React, {Component} from "react";
import {Route, Routes} from "react-router-dom";
import Saying from "../pages/Saying";
import SayList from "../pages/SayList";

class App extends Component {

  render(){

    return (
      <Routes>
        <Route path="/" element={<Saying />} />
        <Route path="/SayList" element={<SayList />} />
      </Routes>
    );

  }

}

export default App;

```

실행해보기

- npm start

index.js

```

import React from 'react';
import ReactDOM from 'react-dom';
import './index.css';
import App from './App';
import { BrowserRouter } from 'react-router-dom';

ReactDOM.render(
  <BrowserRouter>
    <App />
  </BrowserRouter>,
  document.getElementById('root')
);

```

app.js

```

import Web3 from "web3";
import React, {useRef, useState, useEffect} from "react";
import { Link } from "react-router-dom";
import "../Saying.css";
import Abi from "../Abi";

const Saying = () => {

  const [web3, setWeb3] = useState();
  const [account, setAccount] = useState('');
  const [pressStart, setPressStart] = useState(false);

  useEffect(()=>{
    if(typeof window.ethereum !== "undefined"){
      try{
        const web = new Web3(window.ethereum);

```

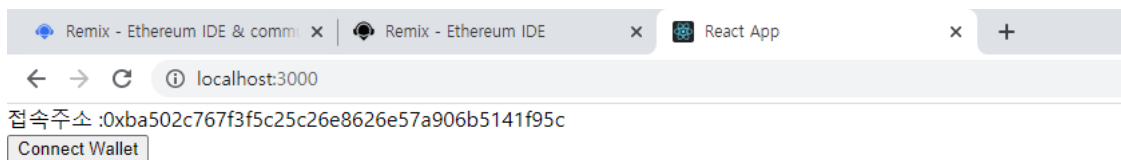
```

        setWeb3(web);
      }catch(err){
        console.log(err);
      }
    }
  }, [])
  //메타마스크연결
  const connectWallet = async()=>{
    const accounts = await window.ethereum.request({
      method : "eth_requestAccounts",
    })
    setAccount(accounts[0]);
  }

  return(
    <div className="app">
      <div className = "account">{pressStart ? "접속주소 : " : ""}{account} </div>
      <button className = "meta"
        onClick={()=>{
          connectWallet();
          setPressStart(true);
        }}>
        Connect Wallet
      </button>
    </div>
  );
}

export default Saying;

```



url("https://images.pexels.com/photos/114979/pexels-photo-114979.jpeg?auto=compress&cs=tinysrgb&dpr=2&w=500");

Saying.css

```

.app{
  display: flex;
  align-items: center;
  justify-content: center;
  height: 100vh;
  background: linear-gradient(
    rgba(255,255,255,0.7),
    rgba(255,255,255,0.3)
  ),
  url("https://images.pexels.com/photos/114979/pexels-photo-114979.jpeg?auto=compress&cs=tinysrgb&dpr=2&w=500");
  background-size: cover;
  background-position: center;
}

button{
  width: 100%;
  height: 50px;
  padding: 10px;
  border: none;
  background-color: rebeccapurple;
  color: white;
  border-radius: 5px;
  font-weight: bold;
  font-size: 18px;
  cursor: pointer;
  margin-top: 15px;
  margin-bottom: 30px;
}

```

```

}

.meta{
  width: 200px;
  position: absolute;
  left: 80%;
  top: 20px;
}

.account{
  position : absolute;
  left : 5%;
  top : 20px;
  font-size : 20px;
  color : gray;
}

```

Saying.js

```

import Web3 from "web3";
import React, {useRef, useState, useEffect} from "react";
import { Link } from "react-router-dom";
import "../Saying.css";
import Abi from "../Abi";

const Saying = () => {

  const [web3, setWeb3] = useState();
  const [account, setAccount] = useState('');
  const [pressStart, setPressStart] = useState(false);
  const [str, setStr] = useState();
  const [ethers, setEthers] = useState();

  //입력
  const [author, setAuthor] = useState("");
  const [saying, setSaying] = useState("");
  const [errorMessage, setErrorMessage] = useState("0.1Eth 수수료가 듭니다.");
  const authorRef = useRef();

  useEffect(() => {
    if(typeof window.ethereum !== "undefined"){
      try{
        const web = new Web3(window.ethereum);
        setWeb3(web);
      }catch(err){
        console.log(err);
      }
    }
  }, [])
  //메타마스크연결
  const connectWallet = async()=>{
    const accounts = await window.ethereum.request({
      method : "eth_requestAccounts",
    })
    setAccount(accounts[0]);
  }

  //이더보내기
  const SendEth = async()=>{
    await web3.eth.sendTransaction({
      form : account,
      to : "0xBa502C767f3F5c25C26e8626e57A906b5141f95C",
      value : web3.utils.toWei("0.1", "ether")
    }).then(function(receipt){
      setEthers(true);
      setStr("구매완료");
    });
  }

  //인풋 함수 입력받은값 블록체인에 함수 호출해서 입력
  const handleSubmit = async(e) =>{
    e.preventDefault(); // 새로고침 안하도록
    console.log(author);
    console.log(saying);

    const ContractAddress = '0x588b0b5ef24bB5Af87bc695B12eA8A4CbF11acEF';
    const Contract = await new web3.eth.Contract(Abi, ContractAddress);

    Contract.methods.addSay(author, saying, false).send({from : account}).on(
      "receipt", (receipt) =>{

```

```

        setStr("명언등록이 성공했습니다.");
    }
}

return(
    <div className="app">
        <div className = "account">{pressStart ? "접속주소 : " : ""}{account} </div>
        <button className = "meta"
            onClick={()=>{
                connectWallet();
                setPressStart(true);
            }}>
            Connect Wallet
        </button>
        <div className="block">
            <h1>블록체인 명언 입력</h1>
            <div className="formInput">
                <label>Author</label>
                <input name="Author" placeholder="Author" onChange={e=>setAuthor(e.target.value)}>
                <label>Saying</label>
                <input name="Saying" placeholder="Saying" onChange={e=>setSaying(e.target.value)}>
            </div>
            {ethers ? <button onClick={handleSubmit}>명언발행</button> : <button onClick={SendEth}> 등록비 구매 완료 후 발행버튼 생성</button>}
            <span>{str}</span>
            <span>{errorMessage}</span>
            <div className = "list">
                <Link to="/SayList">SayList</Link>
            </div>
        </div>
    </div>
);
}

export default Saying;

```

Saying.css

```

.app{
    display: flex;
    align-items: center;
    justify-content: center;
    height: 100vh;
    background: linear-gradient(
        45deg,
        rgba(255,255,255,0.7),
        rgba(255,255,255,0.3)
    ),
    url("https://images.pexels.com/photos/114979/pexels-photo-114979.jpeg?auto=compress&cs=tinysrgb&dpr=2&w=500");
    background-size:cover;
    background-position: center;
}

.formInput{
    display : flex;
    flex-direction: column;
    width : 300px;
}

input{
    padding : 15px;
    margin : 10px 0px;
    border-radius : 5px;
    border : 1px solid gray;
}

.block{
    background-color: white;
    padding : 0px 60px;
    border-radius: 10px;
}

h1{
    color : rgb(71,1,71);
    text-align : center;
}

.list{
    width:100%;
    height:40px;
    padding:10px;
    border:none;
}

```



```

background-color: rgb(34,110,251);
color: white;
border-radius : 5px;
font-weight: bold;
font-size: 18px;
cursor:pointer;
margin-top: 15px;
margin-bottom : 30px;
}

label{
font-size: 12px;
color : gray;
}

span{
font-size:12px;
padding:3px;
color:red;
}

button{
width: 100%;
height: 50px;
padding: 10px;
border: none;
background-color: rebeccapurple;
color: white;
border-radius: 5px;
font-weight: bold;
font-size: 18px;
cursor: pointer;
margin-top: 15px;
margin-bottom: 30px;
}

.meta{
width: 200px;
position:absolute;
left:80%;
top: 20px;
}

.account{
position : absolute;
left : 5%;
top : 20px;
font-size : 20px;
color : gray;
}

```