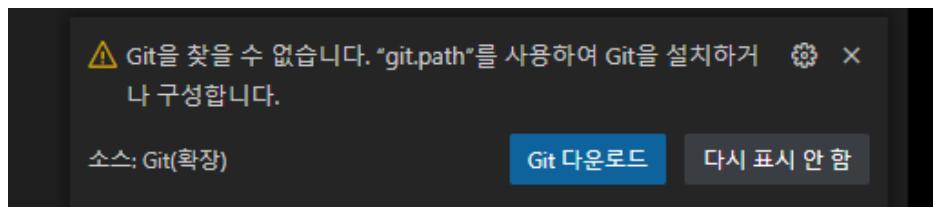


day53-rn-nft-connect

:≡ 태그	
💻 날짜	@2022년 12월 14일



리액트 네이티브로 web3 연결

nft 가져오기

<https://www.genfty.com/>

NFT GENERATOR ONLINE TO MAKE
10K NFT COLLECTION

Genfty.com is the easiest NFT generator to create randomized unique 10k NFT collection, upload to IPFS, make smart contract and mint NFTs in minutes without any code.

MAKE NFT COLLECTION >

Don't know where to start? [Watch tutorial video!](#)

No-Code NFT 1.5 ⚡

10K NFT COLLECTION

make nft collection 들어가기

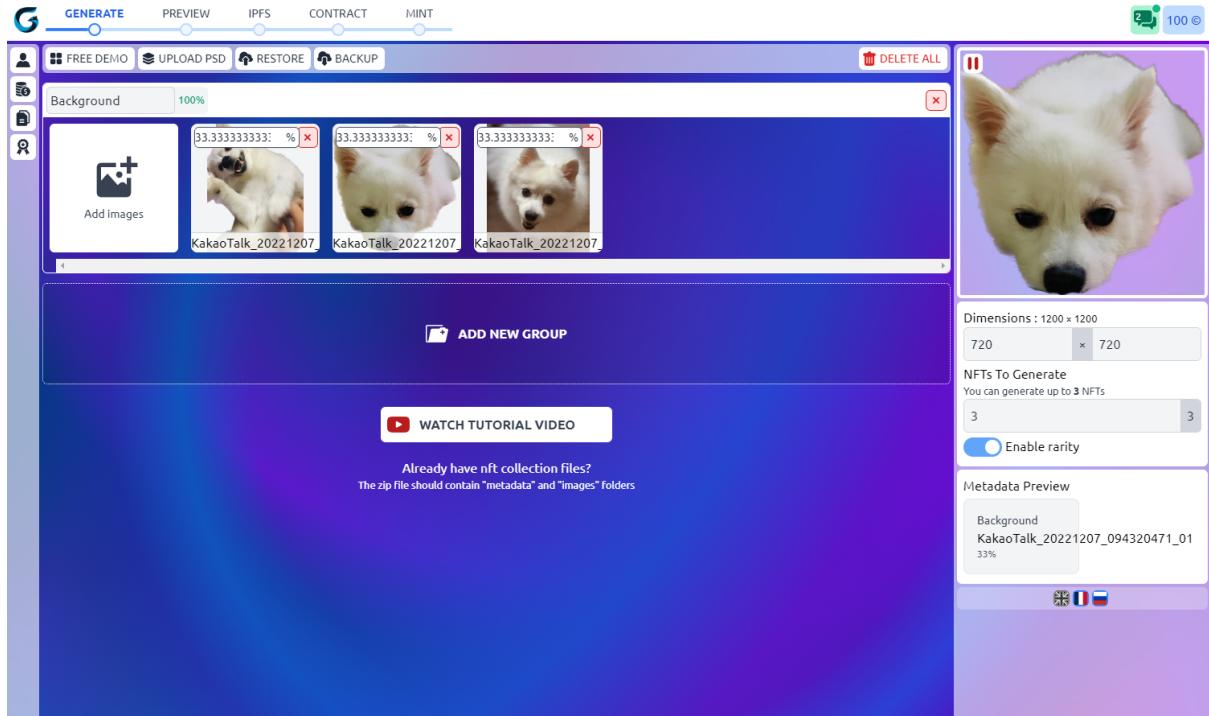
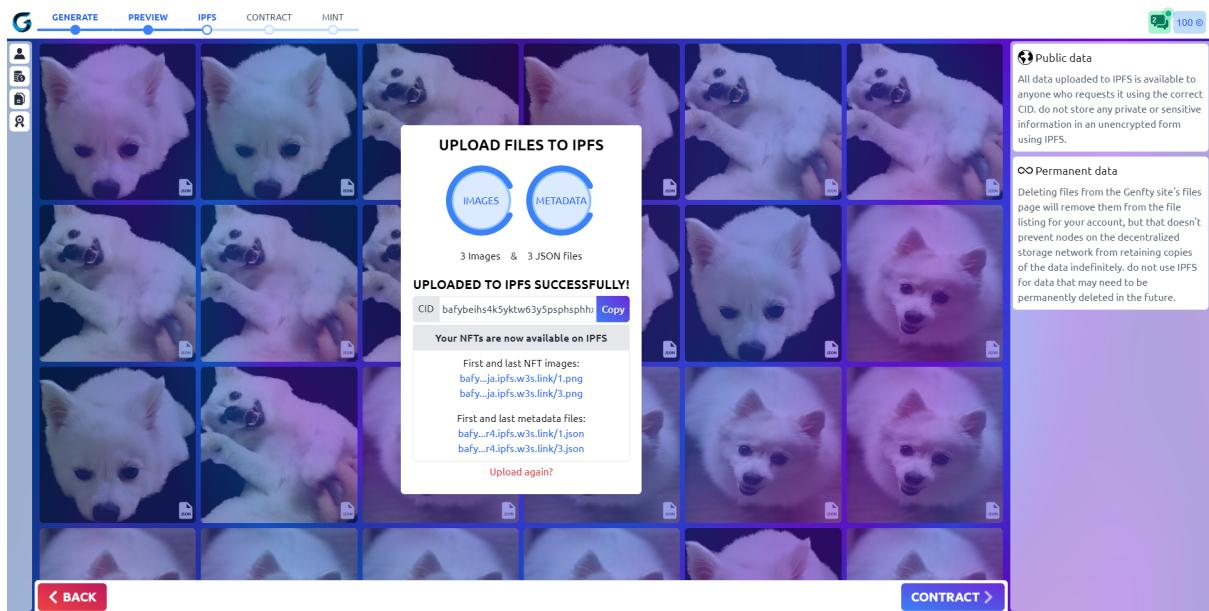


사진 올리기

generate

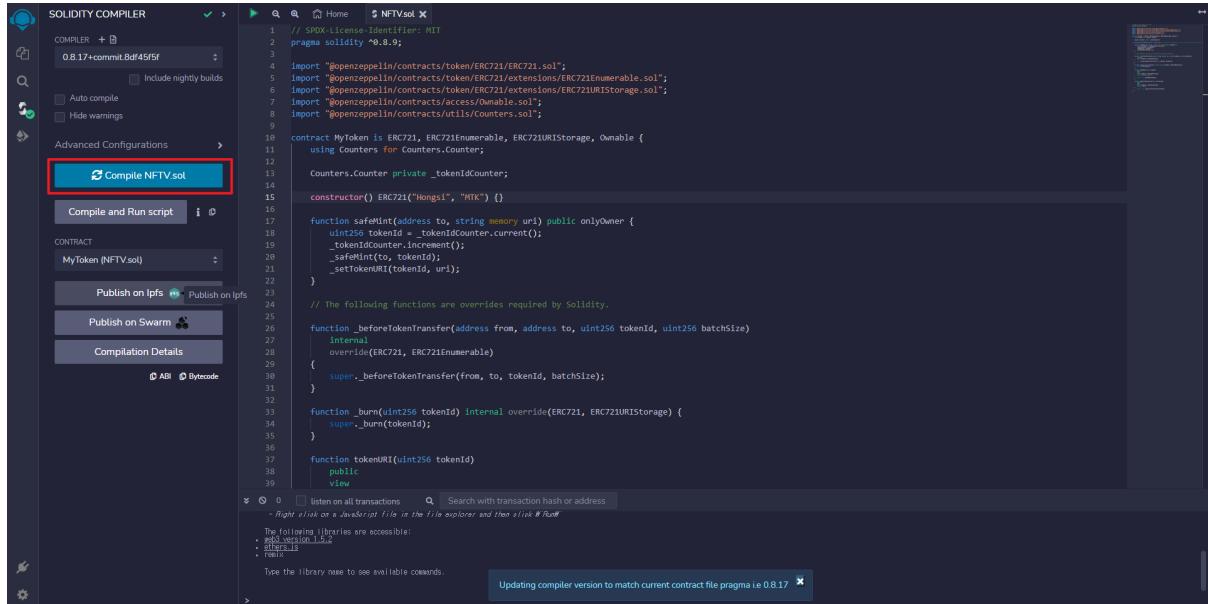
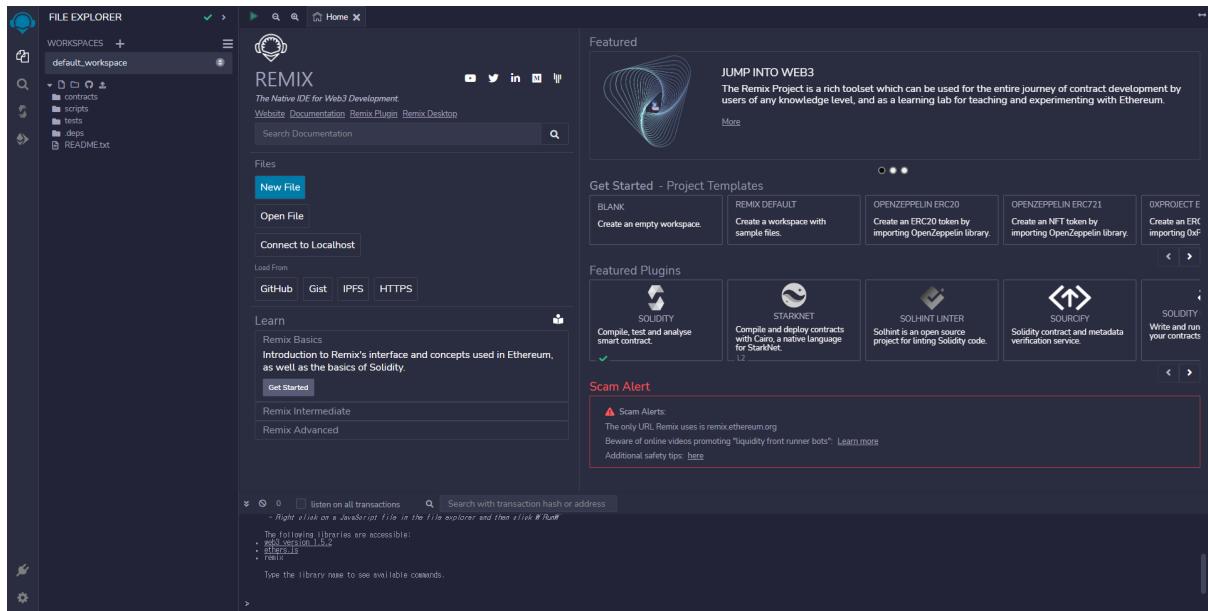


cid : bafybeih4k5yktw63y5psphsphhxbl67m4gxz55d2g4o6brb45zk64bdr4

<https://bafybeidxhv7txar2gtipqeto3msbbgf4bam7tmea3u4lonj4csqo7gmja.ipfs.w3s.link/1.png>

<https://bafybeidxhv7txar2gtipqeto3msbbgf4bam7tmea3u4lonj4csqo7gmja.ipfs.w3s.link/2.png>

<https://bafybeidxhv7txar2gtipqeto3msbbgf4bam7tmea3u4lonj4csqo7gmja.ipfs.w3s.link/3.png>



```

// SPDX-License-Identifier: MIT
pragma solidity ^0.8.9;

import "@openzeppelin/contracts/token/ERC721/ERC721.sol";
import "@openzeppelin/contracts/token/ERC721/extensions/ERC721Enumerable.sol";
import "@openzeppelin/contracts/token/ERC721/extensions/ERC721URIStorage.sol";
import "@openzeppelin/contracts/access/Ownable.sol";
import "@openzeppelin/contracts/utils/Counters.sol";

contract MyToken is ERC721, ERC721Enumerable, ERC721URIStorage, Ownable {
    using Counters for Counters.Counter;

    Counters.Counter private _tokenIdCounter;

    constructor() ERC721("MyToken", "MTK") {}

    function safeMint(address to, string memory uri) public onlyOwner {
        uint256 tokenId = _tokenIdCounter.current();
        _tokenIdCounter.increment();
        _safeMint(to, tokenId);
        _setTokenURI(tokenId, uri);
    }

    // The following functions are overrides required by Solidity.

    function _beforeTokenTransfer(address from, address to, uint256 tokenId, uint256 batchSize)
        internal
        override(ERC721, ERC721Enumerable)
    {
        super._beforeTokenTransfer(from, to, tokenId, batchSize);
    }

    function _burn(uint256 tokenId) internal override(ERC721, ERC721URIStorage) {
        super._burn(tokenId);
    }

    function tokenURI(uint256 tokenId)
        public
        view
        override(ERC721, ERC721URIStorage)
        returns (string memory)
    {
        return super.tokenURI(tokenId);
    }

    function supportsInterface(bytes4 interfaceId)
        public
        view
        override(ERC721, ERC721Enumerable)
        returns (bool)
    {
        return super.supportsInterface(interfaceId);
    }
}

```

Deploy & Run Transactions

ENVIRONMENT: Injected Provider - MetaMask

ACCOUNT: 0xb5...195C (0.207406926)

GAS LIMIT: 300000

VALUE: 0 Wei

CONTRACT (Compiled by Remix): MyToken - contracts/NFTV.sol

Deploy

Publish to IPFS

OR

At Address Load contract from Address

Transactions recorded: 0

Deployed Contracts

Currently you have no contract instances to interact with.

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.9;

import "openzeppelin/contracts/token/ERC721/ERC721.sol";
import "openzeppelin/contracts/token/ERC721/extensions/ERC721Enumerable.sol";
import "openzeppelin/contracts/token/ERC721/extensions/ERC721URIStorage.sol";
import "openzeppelin/contracts/access/Ownable.sol";
import "openzeppelin/contracts/utils/Counters.sol";

contract MyToken is ERC721, ERC721Enumerable, ERC721URIStorage, Ownable {
    using Counters for Counters.Counter;
    Counters.Counter private _tokenIdCounter;

    constructor() ERC721("Hongsi", "HNT") {}

    function safeMint(address to, string memory uri) public onlyOwner {
        uint256 tokenId = _tokenIdCounter.current();
        _tokenIdCounter.increment();
        _safeMint(to, tokenId);
        _setTokenURI(tokenId, uri);
    }

    // The following functions are overrides required by Solidity.

    function _beforeTokenTransfer(address from, address to, uint256 tokenId, uint256 batchSize)
        internal
        override(ERC721, ERC721Enumerable)
    {
        super._beforeTokenTransfer(from, to, tokenId, batchSize);
    }

    function _burn(uint256 tokenId) internal override(ERC721, ERC721URIStorage) {
        super._burn(tokenId);
    }

    function tokenURI(uint256 tokenId)
        public
        view

```

0 Listen on all transactions Search with transaction hash or address

The following libraries are accessible:

- openzeppelin/1.5.9
- stars18/remix

Type the library name to see available commands.

creation of mytoken pending...

Deploy & Run Transactions

ENVIRONMENT: Injected Provider - MetaMask

ACCOUNT: 0xb5...195C (0.207406926)

GAS LIMIT: 300000

VALUE: 0 Wei

CONTRACT (Compiled by Remix): MyToken - contracts/NFTV.sol

Deploy

Publish to IPFS

OR

At Address Load contract from Address

Transactions recorded: 0

Deployed Contracts

Currently you have no contract instances to interact with.

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.9;

import "openzeppelin/contracts/token/ERC721/ERC721.sol";
import "openzeppelin/contracts/token/ERC721/extensions/ERC721Enumerable.sol";
import "openzeppelin/contracts/token/ERC721/extensions/ERC721URIStorage.sol";
import "openzeppelin/contracts/access/Ownable.sol";
import "openzeppelin/contracts/utils/Counters.sol";

contract MyToken is ERC721, ERC721Enumerable, ERC721URIStorage, Ownable {
    using Counters for Counters.Counter;
    Counters.Counter private _tokenIdCounter;

    constructor() ERC721("Hongsi", "HNT") {}

    function safeMint(address to, string memory uri) public onlyOwner {
        uint256 tokenId = _tokenIdCounter.current();
        _tokenIdCounter.increment();
        _safeMint(to, tokenId);
        _setTokenURI(tokenId, uri);
    }

    // The following Functions are overrides required by Solidity.

    function _beforeTokenTransfer(address from, address to, uint256 tokenId, uint256 batchSize)
        internal
        override(ERC721, ERC721Enumerable)
    {
        super._beforeTokenTransfer(from, to, tokenId, batchSize);
    }

    function _burn(uint256 tokenId) internal override(ERC721, ERC721URIStorage) {
        super._burn(tokenId);
    }

    function tokenURI(uint256 tokenId)
        public
        view

```

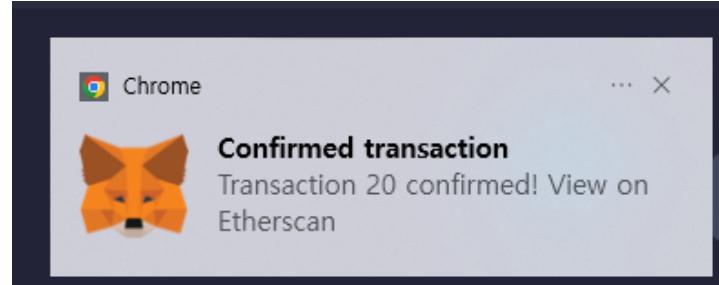
0 Listen on all transactions Search with transaction hash or address

The following libraries are accessible:

- openzeppelin/1.5.9
- stars18/remix

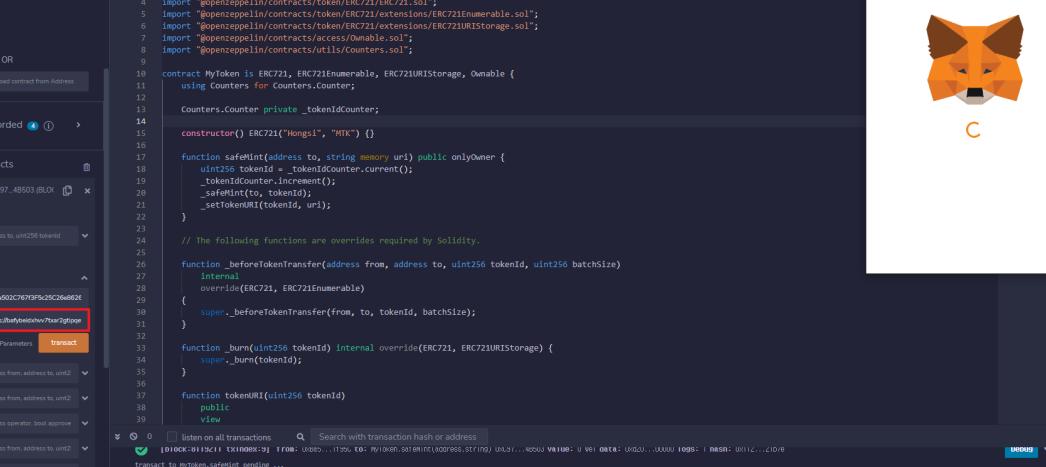
Type the library name to see available commands.

creation of mytoken pending...



The screenshot shows the Truffle UI interface with the following details:

- Deploy & Run Transactions**: Shows a deployment log for "MyToken - contracts/NFTV.sol".
 - Deployment address: 0x0b5...f95C
 - Balance: 0 ETH
- Transactions recorded**: A list of recent transactions, with one highlighted:
 - Approve: address to: uint256 tokenId
 - renounce...
 - safeMint: address to: string
- Deployed Contracts**: A list of deployed contracts, with one highlighted:
 - safeTransferFrom: address from, address to, uint2
 - safeTransferFrom: address from, address to, uint2
 - setApprovalForAll: address operator, bool approve
 - transferFrom: address from, address to, uint2
 - transferFromO: address newOwner
 - balanceOf: address owner
 - getApproved: uint256 tokenId
- Code View**: The Solidity code for the `MyToken` contract, showing functions like `approve`, `safeMint`, and various `safeTransferFrom` variants.
- Account**: Shows the account balance in GoerliETH.
 - 0.1993 GoerliETH (0x0b5...f95C)
 - 999999000 HOO
- Bottom Bar**: Includes buttons for "Debug" and "Deploy".



The screenshot shows a Solidity development environment with the following details:

- Deploy & Run Transactions** tab is active.
- MyToken - contracts/NFTv1.sol** is the current file.
- Deploy** button is visible.
- Publish to IPFS** and **OR** options are present.
- At Address** and **Load contract from Address** buttons are available.
- Transactions recorded**: 4 pending.
- Deployed Contracts** section shows:
 - MYTOKEN At 0x9C7...4B503 (BLOCK 1)**:
 - Balance: 0 ETH
 - Approve**: address to: 0x00
 - Renounce**:
 - safeMint**:
 - to: 0x6a502c7679f9c25c26e8902
 - URI**: <https://nfttest.mew7.us/2gooe>
 - safeTransfer**: address from, address to, uint256
 - safeTransferFrom**: address from, address to, uint256
 - setApprovalForAll**: address operator, bool approve
 - transferFrom**: address from, address to, uint256
 - transferOwnership**: address newOwner
 - balanceOf**: address owner
 - getApproved**: uint256 tokenID
- Search bar** at the top right: `0 / listen on all transactions` and `Search with transaction hash or address`.
- Recent Transactions** table:

ID	Block	Gas used	Gas price	From	To	Value	Data	Fees	Hash
1	0x112...	0x112...	0x112...	0x00	0x00	0x00	0x00	0x00	0x00
- Bottom right corner**: **DEPLOY** button.

The screenshot shows the Etherscan interface for a transaction. The transaction details are as follows:

- tokenOfOwner**: address owner, uint256 index
- tokenURI**: tokenid: 0
- call**: tokenURI - call
- call to**: MyToken.tokenURI
- CALL [call] from**: 0x8502C76f13F5c2526e8626e57A906b514f195C to: MyToken.tokenURI(uint256) data: 0xc87...00000

The "call" button is highlighted in blue. The "Parameters" tab is also visible.

The top screenshot shows the tokenURI function for tokenID 1. The bottom screenshot shows the tokenURI function for tokenID 2. Both screenshots show the same URI: https://ipfs.w3s.link/2.png.

컨트렉트

0xC97423BA5B77Ab3D3D4E21437FAb33701C14B503

abi

```
[
  {
    "inputs": [],
    "stateMutability": "nonpayable",
    "type": "constructor"
  },
  {
    "anonymous": false,
    "inputs": [
      {
        "indexed": true,
        "internalType": "address",
        "name": "owner",
        "type": "address"
      },
      {
        "indexed": true,
        "internalType": "address",
        "name": "approved",
        "type": "address"
      },
      {
        "indexed": true,
        "internalType": "uint256",
        "name": "tokenId",
        "type": "uint256"
      }
    ],
    "name": "Approval",
    "type": "event"
  }
]
```

```

},
{
  "anonymous": false,
  "inputs": [
    {
      "indexed": true,
      "internalType": "address",
      "name": "owner",
      "type": "address"
    },
    {
      "indexed": true,
      "internalType": "address",
      "name": "operator",
      "type": "address"
    },
    {
      "indexed": false,
      "internalType": "bool",
      "name": "approved",
      "type": "bool"
    }
  ],
  "name": "ApprovalForAll",
  "type": "event"
},
{
  "inputs": [
    {
      "internalType": "address",
      "name": "to",
      "type": "address"
    },
    {
      "internalType": "uint256",
      "name": "tokenId",
      "type": "uint256"
    }
  ],
  "name": "approve",
  "outputs": [],
  "stateMutability": "nonpayable",
  "type": "function"
},
{
  "anonymous": false,
  "inputs": [
    {
      "indexed": true,
      "internalType": "address",
      "name": "previousOwner",
      "type": "address"
    },
    {
      "indexed": true,
      "internalType": "address",
      "name": "newOwner",
      "type": "address"
    }
  ],
  "name": "OwnershipTransferred",
  "type": "event"
},
{

```

```
"inputs": [],
  "name": "renounceOwnership",
  "outputs": [],
  "stateMutability": "nonpayable",
  "type": "function"
},
{
  "inputs": [
    {
      "internalType": "address",
      "name": "to",
      "type": "address"
    },
    {
      "internalType": "string",
      "name": "uri",
      "type": "string"
    }
  ],
  "name": "safeMint",
  "outputs": [],
  "stateMutability": "nonpayable",
  "type": "function"
},
{
  "inputs": [
    {
      "internalType": "address",
      "name": "from",
      "type": "address"
    },
    {
      "internalType": "address",
      "name": "to",
      "type": "address"
    },
    {
      "internalType": "uint256",
      "name": "tokenId",
      "type": "uint256"
    }
  ],
  "name": "safeTransferFrom",
  "outputs": [],
  "stateMutability": "nonpayable",
  "type": "function"
},
{
  "inputs": [
    {
      "internalType": "address",
      "name": "from",
      "type": "address"
    },
    {
      "internalType": "address",
      "name": "to",
      "type": "address"
    },
    {
      "internalType": "uint256",
      "name": "tokenId",
      "type": "uint256"
    }
  ],
  "name": "transferFrom",
  "outputs": [],
  "stateMutability": "nonpayable",
  "type": "function"
}
```

```

        {
          "internalType": "bytes",
          "name": "data",
          "type": "bytes"
        }
      ],
      "name": "safeTransferFrom",
      "outputs": [],
      "stateMutability": "nonpayable",
      "type": "function"
    },
    {
      "inputs": [
        {
          "internalType": "address",
          "name": "operator",
          "type": "address"
        },
        {
          "internalType": "bool",
          "name": "approved",
          "type": "bool"
        }
      ],
      "name": "setApprovalForAll",
      "outputs": [],
      "stateMutability": "nonpayable",
      "type": "function"
    },
    {
      "anonymous": false,
      "inputs": [
        {
          "indexed": true,
          "internalType": "address",
          "name": "from",
          "type": "address"
        },
        {
          "indexed": true,
          "internalType": "address",
          "name": "to",
          "type": "address"
        },
        {
          "indexed": true,
          "internalType": "uint256",
          "name": "tokenId",
          "type": "uint256"
        }
      ],
      "name": "Transfer",
      "type": "event"
    },
    {
      "inputs": [
        {
          "internalType": "address",
          "name": "from",
          "type": "address"
        },
        {
          "internalType": "address",
          "name": "to",
          "type": "address"
        }
      ]
    }
  ]
}

```

```

        "type": "address"
    },
    {
        "internalType": "uint256",
        "name": "tokenId",
        "type": "uint256"
    }
],
"name": "transferFrom",
"outputs": [],
"stateMutability": "nonpayable",
"type": "function"
},
{
    "inputs": [
        {
            "internalType": "address",
            "name": "newOwner",
            "type": "address"
        }
    ],
"name": "transferOwnership",
"outputs": [],
"stateMutability": "nonpayable",
"type": "function"
},
{
    "inputs": [
        {
            "internalType": "address",
            "name": "owner",
            "type": "address"
        }
    ],
"name": "balanceOf",
"outputs": [
    {
        "internalType": "uint256",
        "name": "",
        "type": "uint256"
    }
],
"stateMutability": "view",
"type": "function"
},
{
    "inputs": [
        {
            "internalType": "uint256",
            "name": "tokenId",
            "type": "uint256"
        }
    ],
"name": "getApproved",
"outputs": [
    {
        "internalType": "address",
        "name": "",
        "type": "address"
    }
],
"stateMutability": "view",
"type": "function"
},

```

```
{
  "inputs": [
    {
      "internalType": "address",
      "name": "owner",
      "type": "address"
    },
    {
      "internalType": "address",
      "name": "operator",
      "type": "address"
    }
  ],
  "name": "isApprovedForAll",
  "outputs": [
    {
      "internalType": "bool",
      "name": "",
      "type": "bool"
    }
  ],
  "stateMutability": "view",
  "type": "function"
},
{
  "inputs": [],
  "name": "name",
  "outputs": [
    {
      "internalType": "string",
      "name": "",
      "type": "string"
    }
  ],
  "stateMutability": "view",
  "type": "function"
},
{
  "inputs": [],
  "name": "owner",
  "outputs": [
    {
      "internalType": "address",
      "name": "",
      "type": "address"
    }
  ],
  "stateMutability": "view",
  "type": "function"
},
{
  "inputs": [
    {
      "internalType": "uint256",
      "name": "tokenId",
      "type": "uint256"
    }
  ],
  "name": "ownerOf",
  "outputs": [
    {
      "internalType": "address",
      "name": "",
      "type": "address"
    }
  ]
}
```

```
        },
      ],
      "stateMutability": "view",
      "type": "function"
    },
    {
      "inputs": [
        {
          "internalType": "bytes4",
          "name": "interfaceId",
          "type": "bytes4"
        }
      ],
      "name": "supportsInterface",
      "outputs": [
        {
          "internalType": "bool",
          "name": "",
          "type": "bool"
        }
      ],
      "stateMutability": "view",
      "type": "function"
    },
    {
      "inputs": [],
      "name": "symbol",
      "outputs": [
        {
          "internalType": "string",
          "name": "",
          "type": "string"
        }
      ],
      "stateMutability": "view",
      "type": "function"
    },
    {
      "inputs": [
        {
          "internalType": "uint256",
          "name": "index",
          "type": "uint256"
        }
      ],
      "name": "tokenByIndex",
      "outputs": [
        {
          "internalType": "uint256",
          "name": "",
          "type": "uint256"
        }
      ],
      "stateMutability": "view",
      "type": "function"
    },
    {
      "inputs": [
        {
          "internalType": "address",
          "name": "owner",
          "type": "address"
        }
      ],
      {
```

```
        "internalType": "uint256",
        "name": "index",
        "type": "uint256"
    },
],
"name": "tokenOfOwnerByIndex",
"outputs": [
{
    "internalType": "uint256",
    "name": "",
    "type": "uint256"
}
],
"stateMutability": "view",
"type": "function"
},
{
"inputs": [
{
    "internalType": "uint256",
    "name": "tokenId",
    "type": "uint256"
}
],
"name": "tokenURI",
"outputs": [
{
    "internalType": "string",
    "name": "",
    "type": "string"
}
],
"stateMutability": "view",
"type": "function"
},
{
"inputs": [],
"name": "totalSupply",
"outputs": [
{
    "internalType": "uint256",
    "name": "",
    "type": "uint256"
}
],
"stateMutability": "view",
"type": "function"
}
]
```

VS Code screenshot showing the file structure of an Expo project. The 'erc721Abi.js' file is selected and highlighted with a red box.

```

const erc721Abi = [
  {
    "inputs": [],
    "stateMutability": "nonpayable",
    "type": "constructor"
  },
  {
    "anonymous": false,
    "inputs": [
      {
        "indexed": true,
        "internalType": "address",
        "name": "owner",
        "type": "address"
      },
      {
        "indexed": true,
        "internalType": "address",
        "name": "approved",
        "type": "address"
      }
    ]
  }
]

```

abi 넣기

```

54
55   export default erc721Abi;

```

VS Code screenshot showing the 'App.js' file where the ABI is being imported and used to define the App component.

```

import { StatusBar } from 'expo-status-bar';
import { StyleSheet, Text, View } from 'react-native';
import './global';
import Web3 from 'web3';
import { useEffect, useState } from 'react';
import erc721Abi from './erc721Abi';

export default function App() {
  // 자신의 계정 잔고 조회
  const [web3, setWeb3] = useState();
  const [balance, setBalance] = useState();
  const [account, setAccount] = useState('');
  const [erc721address, setErc721address] = useState();
  const [erc721list, setErc721list] = useState([]);

  // 초기에 한번 함수동작
  useEffect(()=>{
    ...
  }, []);
}

```

```
● JS erc721Abi.js
5 > ⚙️ App > 🐾 addErc721Token
//동기화 async await
const getBalance = async ()=>{
  let balance = await web3.eth.getBalance("0x6e51918a5724Bd25E7013DeB9dD23043515117");
  balance = web3.utils.fromWei(balance, 'ether');
  console.log(balance);
  setBalance(balance);
}

const addErc721Token = async () => {
  // 내 메타마스크 주소
  setAccount('0xBA502C767f3F5c25C26e8626e57A906b5141f95C');

  //컨트랙트 주소
  setErc721address('0xC97423BA5B77Ab3D3D4E21437FAb33701C14B503');
  //컨트랙트 주소와 ABI 필요
  const tokenContract = await new web3.eth.Contract(erc721Abi, erc721address);

  console.log(tokenContract);

  const name = await tokenContract.method.name().call();
  const symbol = await tokenContract.methods.symbol().call();
  const totalSupply = await tokenContract.methods.totalSupply().call();

  let arr = [];
  for(let i = 0; i<totalSupply; i++){
    arr.push(i);
  }

  for(let tokenId of arr){
    const tokenURI = await tokenContract.methods.tokenURI(tokenId).call();
    console.log("tokenURI : " + tokenURI);

    setErc721address((prev)=> {
      return [...prev, {name, symbol, tokenId, tokenURI}];
    })
  }
}
```

이거 주소 넣어야한다 useEffect 부분

The screenshot shows the Infura API endpoint configuration interface. On the left, there's a sidebar with icons for Stats, DNS, and IPFS. The main area has tabs for ENDPOINTS, SETTINGS, SECURITY, and PROJECT SHARING, with ENDPOINTS being the active tab.

GET STARTED WITH WEB3

- Ethereum** L1 Blockchain: GÖERLI dropdown, URL: <https://goerli.infura.io>
- Optimism** Ethereum L2/Rollup: select dropdown, URL: <https://optimism-mainnet.infura.io>
- Polygon** Ethereum Sidechain: select dropdown, URL: <https://polygon-mumbai.infura.io>
- Arbitrum** Ethereum L2/Rollup: select dropdown, URL: <https://arbitrum-one.infura.io>
- Avalanche C-Chain** Ethereum L1: MAINNET dropdown, URL: <https://avalanche-mainnet.infura.io>
- NEAR** PoS L1: MAINNET dropdown, URL: <https://near-mainnet.infura.io>
- Palm** Ethereum NFT Sidechain: MAINNET dropdown, URL: <https://palm-mainnet.infura.io>
- Starknet** Ethereum ZK-Rollup L2: MAINNET dropdown, URL: <https://starknet-mainnet.infura.io>

0 TO NFT IN JUST A FEW MINUTES

NFT API Higher Level NFT API: The NFT API sends REST calls to multiple networks to retrieve NFT data. It can be used with your other endpoints and the NFT SDK library to deploy and call methods on common Ethereum smart-contract definitions through Infura without the developer overhead of learning Solidity, compiling code, importing ABI's, etc.

Getting Started

YOUR FIRST REQUEST

Retrieve the current block number
Copy and paste the code below into Terminal and your response should be the current block number.

```
curl --url https://mainnet.infura.io/v3/a303bfe01e1a4\n  \n-X POST \n-H "Content-Type: application/json" \n-d '{"jsonrpc":"2.0","method":"eth_blockNumber"}' \n
```

Copy

Your response should look like:

```
{"jsonrpc":"2.0","id":1,"result":"0x8d8abf"}\n
```

PROJECT SECURITY

GETTING YOUR PROJECT STARTED

VIEW YOUR PROJECT STATS

GETTING HELP

OTHER RESOURCES

