

블록체인 기술 _ 이더리움 2주차 보고서

항공전자정보공학부 2020124184 차정은

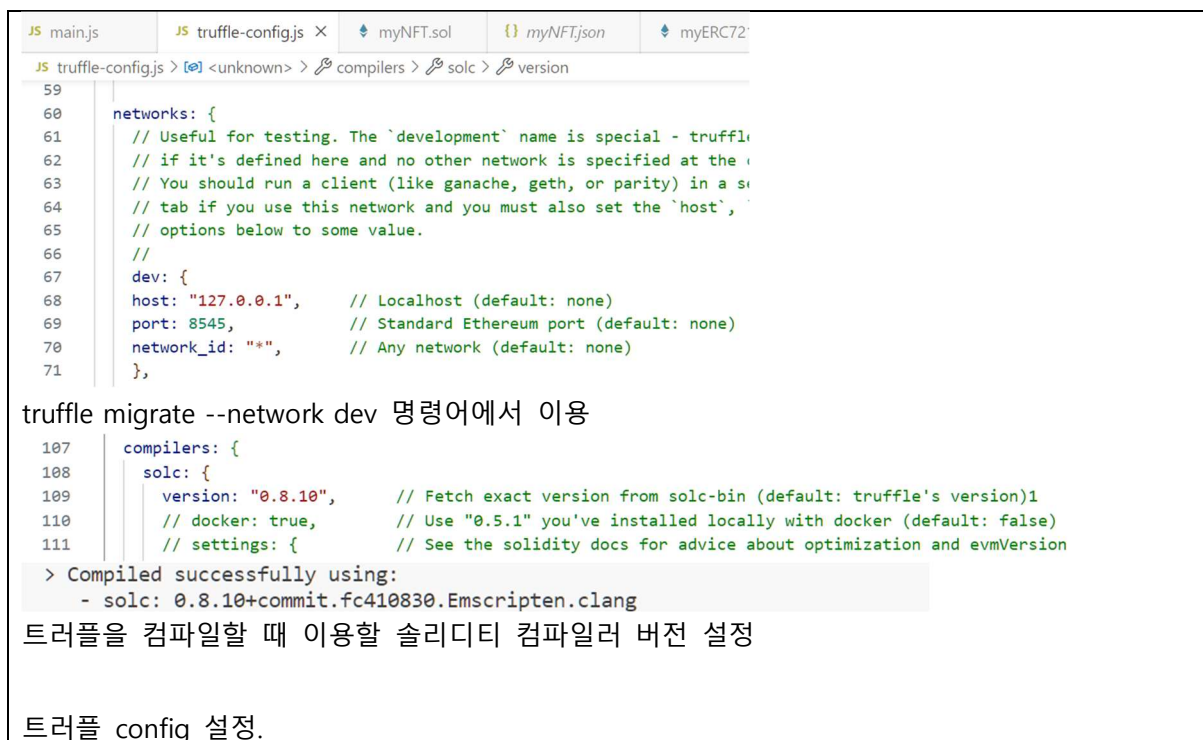
0. 기본 세팅



```
ipfs config --json API.HTTPHeaders.Access-Control-Allow-Origin '["*"]' 해주기 위함
```

명령 프롬프트와 power shell에 입력했을 때 수정이 되지 않아 직접 config 파일 수정

```
C:\Users\cygnu\OneDrive\바탕 화면\kubo>
C:\Users\cygnu\OneDrive\바탕 화면\kubo>ipfs config --json API.HTTPHeaders.Access-Control-Allow-Origin '["*"]'
Error: failed to unmarshal json: invalid character 'W' looking for beginning of value
```



1. myNFT.sol

```
1  // SPDX-License-Identifier: UNLICENSED
2  pragma solidity ^0.8.10;
3
4  import "./myERC721.sol";
5
6  contract myNFT is ERC721 {
7      // ERC721 기능 구현
8
9      /* 기능 구현을 위한 mapping */
10     /* 주소가 소유한 NFT 개수 저장 */
11     mapping(address => uint256) public tokenBalance;
12     /* 토큰Id를 소유한 주소 저장 */
13     mapping(uint256 => address) public tokenOwner;
14     /* 토큰Id의 가격 저장 (판매중일 땐 5 ETH, 아니면 0) */
15     mapping(uint256 => uint256) public tokenPrice;
16     /* 토큰Id의 판매 여부 저장 */
17     mapping(uint256 => bool) public isForSale;
18     /* 토큰Id의 소유자 주소 내역 저장 */
19     mapping(uint256 => address[]) public tokenOwnerHistory;
20
21     // NFT 정의
22     struct Image {
23         string name;
24         string url;
25         string myAttributes;
26         /* 원하는 만큼 속성을 정의 및 추가하셔도 됩니다. */
27     }
28     Image[] public images;
29
30     constructor(
31         string memory _name,
32         string memory _symbol
33     ) ERC721({}) {}
34
35     function mint(
36         address _to,    // NFT를 발행하고 소유할 주소
37         string memory _name,    // NFT의 이름
38         string memory _url,    // NFT의 path
39         string memory myAttributes    // NFT의 속성
40     ) public {
41         /* NFT를 소유하는 주소는 address(0)가 아니어야 함 */
42         require(_to != address(0), "Empty address cannot mint NFT");
43         uint _tokenId = images.length; /* 토큰ID는 발행될 때마다 0부터 차례대로 부여 */
44         require(!_exists(_tokenId), "1 tokenId can only be assigned to 1 NFT");
45         /* 발행된 토큰Id를 서로 다른 두 NFT가 공유해서는 안 됨 */
46         images.push( /* images 배열에 발행된 NFT 추가 */
47             Image(_name, _url, myAttributes)
48         );
49
50         tokenOwnerHistory[_tokenId].push(_to);
51         /* 발행된 NFT의 소유주 기록에 발행자의 주소 추가 */
52         _mint(_to, _tokenId);
53     }
```

```

54 // nft 발행 기능
55 function _mint(address to, uint256 tokenId) internal {
56     tokenOwner[tokenId] = to;
57     tokenBalance[to] += 1;
58     emit Transfer(address(0), to, tokenId);
59 }
60
61 // tokenId의 소유자 기록 반환
62 function ownerHistoryOf(uint256 tokenId) public
63     view returns (address[] memory) {
64     return tokenOwnerHistory[tokenId];
65 }
66
67 // NFT id 생성 여부 확인
68 function _exists(uint256 tokenId) internal
69     view returns (bool) {
70     /* tokenId의 소유자를 받아옵니다. */
71     address owner = tokenOwner[tokenId];
72     return owner != address(0);
73 }
74
75 // 토큰의 소유자 계정 찾기
76 function ownerOf(uint256 _tokenId) public
77     view returns (address) {
78     return tokenOwner[_tokenId];
79 }
80
81 // 토큰 판매 시작
82 function sellToken(uint256 _tokenId) public {
83     /* 함수를 호출하는 주소는 해당 토큰Id의 보유자여야 합니다. */
84     require(msg.sender == tokenOwner[_tokenId], "Only owner can sell token");
85     /* onSale 함수를 호출해 해당 토큰Id의 판매 상태를 true로 바꿉니다. */
86     onSale(_tokenId, true);
87     /* 토큰의 가격은 5 ETH로 고정합니다. */
88     tokenPrice[_tokenId] = 5 ether;
89 }
90
91 // 토큰 판매 여부 설정
92 function onSale(uint256 _tokenId, bool _forSale) public {
93     isForSale[_tokenId] = _forSale;
94 }
95
96 // 토큰 구매 기능
97 function buyToken(uint256 _tokenId) public payable {
98     /*
99     유효성 확인:
100     1 토큰이 판매중인가
101     2 전송한 이더량이 토큰을 살 수 있는 양인가
102     3 판매자와 구매자가 동일 인물이 아닌가
103     */
104     require(isOnSale(_tokenId), "Token is not on sale");
105     require(msg.value >= tokenPrice[_tokenId], "Not enough ETH");
106     address owner = ownerOf(_tokenId);
107     require(msg.sender != owner, "You cannot buy your own token");
108
109     /* 토큰의 가격만큼의 ETH를 판매자에게 전송 */
110     payable(owner).transfer(tokenPrice[_tokenId]);

```

```

117     /* 토큰을 구매자의 소유로 이동 */
118     transferFrom(owner, msg.sender, _tokenId);
119     /* 토큰 소유주 내역에 구매자의 주소 추가 */
120     tokenOwnerHistory[_tokenId].push(msg.sender);
121     /* 토큰 가격을 지불하고 남은 금액을 구매자에게 재전송 */
122     payable(msg.sender).transfer(msg.value - tokenPrice[_tokenId]);
123
124     /* 토큰의 판매 여부 초기화 */
125     isForSale[_tokenId] = false;
126     tokenPrice[_tokenId] = 0;
127 }
128
129 // 토큰 전송 기능
130 function transferFrom(
131     address _from,
132     address _to,
133     uint256 _tokenId
134 ) public {
135     /*
136     | 전송하고자 하는 토큰Id의 소유자 주소를 가져와 유효성 검사:
137     | 1 판매자가 빈 주소가 아니어야 하며 토큰Id의 소유자여야 함
138     | 2 구매자가 빈 주소가 아니어야 함
139     */
140     address owner = tokenOwner[_tokenId];
141     require(_from != address(0), "Validation Failed: Seller must not be empty address");
142     require(owner == _from, "Validation Failed: Seller must be owner");
143     require(_to != address(0), "Validation Failed: Buyer must not be empty address");
144     /*
145     | 각 주소가 보유한 토큰량을 가감하고
146     | 토큰Id의 현 소유자 주소 변경
147     */
148     tokenBalance[_from] -= 1;
149     tokenOwner[_tokenId] = address(0); // 소유자 주소 변경 전 초기화
150     tokenBalance[_to] += 1;
151     tokenOwner[_tokenId] = _to; // 새 소유자 주소로 업데이트
152     /* 토큰 전송 이벤트 호출 */
153     emit Transfer(owner, _to, _tokenId);
154 }
155
156 // 계좌 내 토큰 개수 확인
157 function balanceOf(address _owner) public
158     view override returns (uint256) {
159     return tokenBalance[_owner];
160 }
161 }

```


2. migration

```
1  const ImageNFT = artifacts.require("myNFT");
2
3  module.exports =function (deployer){
4      const name="MyNFT";
5      const symbol="MNT";
6      deployer.deploy(ImageNFT,name,symbol);
7  }
```

네트워크에 컨트랙트를 배포할 수 있는 js 코드

```
PS C:\Users\cygnu\Downloads\Practice_2> truffle compile
```

● Compiling your contracts...

=====

```
> Compiling .\contracts\myERC721.sol
> Compiling .\contracts\myNFT.sol
> Artifacts written to C:\Users\cygnu\Downloads\Practice_2\build\contracts
> Compiled successfully using:
  - solc: 0.8.10+commit.fc410830.Emscripten.clang
```

```
PS C:\Users\cygnu\Downloads\Practice_2> truffle migrate --network dev
```

● Compiling your contracts...

=====

```
> Compiling .\contracts\myERC721.sol
> Compiling .\contracts\myNFT.sol
> Artifacts written to C:\Users\cygnu\Downloads\Practice_2\build\contracts
> Compiled successfully using:
  - solc: 0.8.10+commit.fc410830.Emscripten.clang
```

Starting migrations...

=====

```
> Network name:      'dev'
> Network id:        1686569466316
> Block gas limit: 6721975 (0x6691b7)
```

1_mynft_migration.js

=====

Deploying 'myNFT'

```
> transaction hash:  0xcd9c6e1e5049f7b2f63ef5e4065ea2e1ce19557c5802c9b36b9109d848656396
> Blocks: 0          Seconds: 0
> contract address:  0x0CE6522A0aa0B9D5F06903a53a6377b394eB2Aae
> block number:      1
> block timestamp:    1686569504
> account:            0xf6ab0381201FE3f9727C52BE4D00812e62717385
> balance:            99.96476986
> gas used:           1761507 (0x1ae0e3)
> gas price:          20 gwei
> value sent:         0 ETH
> total cost:         0.03523014 ETH
```

> Saving artifacts

```
> Total cost:         0.03523014 ETH
```

Summary

=====

```
> Total deployments:  1
> Final cost:         0.03523014 ETH
```

트러플 컴파일, migration

```
JS main.js    {} myNFT.json X
build > contracts > {} myNFT.json > {} ast > [ ] nodes > {} 2
1  {
2    "contractName": "myNFT",
3  > "abi": [ ...
406 ],
407 "metadata": "{\\"compiler\\":{\\"version
...
Deploying 'myNFT'
-----
> transaction hash:    0xcd9c6e1e5049f7b2f63ef5e4065ea2e1ce19557c!
> Blocks: 0           Seconds: 0
> contract address:   0x0CE6522A0aa0B9D5F06903a53a6377b394eB2Aae

JS main.js    X
JS main.js > [ABI] > "inputs"
1  /* Ganache 및 IPFS와 연결 */
2  const web3 = new Web3(new Web3.providers.HttpProvider("http://localhost:8545"));
3  const ipfs = window.IpfsHttpClient.create("http://127.0.0.1:5001");
4
5  /* 컨트랙트 배포 후 컨트랙트 주소와 ABI 입력 */
6  const CA = "0x0CE6522A0aa0B9D5F06903a53a6377b394eB2Aae";
7  const ABI = [{
8    "inputs": [
9      {
10       "internalType": "string",
11       "name": "_name",
12       "type": "string"
13     },

```

컴파일 및 mygration으로 생성된 CA와 ABI를 main.js에 삽입

3. 실행 결과

← → ↺ 127.0.0.1:5500/main.html

지갑 주소: 0xf6ab0381201FE3f9727C52BE4D00812e62717385

보유 ETH: 99.96476986

보유 NFT: 0

0x365ed54817828fc3fdb00e

지갑 가져오기

가나슈에 생성된 지갑의 프라이빗키를 넣어 확인

Name je

File
파일 선택 he-junhui-7v...A-unsplash.jpg

My Attributes Value1 ▾

NFT 생성하기

127.0.0.1:5500 내용:
NFT 생성 완료

확인

콘솔 새로운 기능 문제

top 필터 기본 수준 문제 15건: 15

Pk inputted: main.js:421
0x365ed54817828fc3fdb00e41f049e56a74efe7f3c804767963028eefedc9b967

wallet import with private key main.js:432

Image CID: QmTbS2HvLnvjkPP4gvVKLoot5Y3uquLYpLg7F2CkT6Wekc main.js:455

Metadata CID: QmV3jAAN1LA5u1nhVXxwaN1Qvre8raAiNqb6z1GT2q7c73 main.js:456

Image Metadata: main.js:457

mintingAddress : 0xf6ab0381201FE3f9727C52BE4D00812e62717385
ipfsPath : QmTbS2HvLnvjkPP4gvVKLoot5Y3uquLYpLg7F2CkT6Wekc

name : je
nftAttribute1 : value1

> Transaction: 0xceac6476a0a4d6e91a03c2aead25d9d8c5835ad8d26b629c5a101f39689efef8
Gas usage: 238760
Block Number: 2
Block Time: Mon Jun 12 2023 22:03:05 GMT+0900 (대한민국 표준시)

그림을 넣어 NFT 생성

지갑 주소: 0xf6ab0381201FE3f9727C52BE4D00812e62717385

보유 ETH: 99.955370932

보유 NFT: 2

0x365ed54817828fc3fdb00e

지갑 가져오기

Name 자연

Image CID: QmRYRwedf9NwsVMin5k3N1MUESggCXBPCN64TrgNEjLayM	main.js:455
Metadata CID: QmZkgqrNK9smsgzGYLt2o9cobBDrjUrFAMKwTcMSqCX2DS	main.js:456
Image Metadata:	main.js:457
mintingAddress : 0xf6ab0381201FE3f9727C52BE4D00812e62717385 ipfsPath : QmRYRwedf9NwsVMin5k3N1MUESggCXBPCN64TrgNEjLayM name : 자연 nftAttribute1 : value1	
NFT minted	main.js:477
> Transaction: 0xa677b42ed93ee9eeb126977265ece5eb83f9014e567d350c75ebade5ca75cdb0 Gas usage: 208808 Block Number: 3 Block Time: Mon Jun 12 2023 22:06:52 GMT+0900 (대한민국 표준시)	
그림을 넣어 NFT 생성	

토큰 ID

0

Owner address 0xf6ab0381201FE3f9727C52BE4D00812e62717385

Name je

File path QmTbS2HvLnvjkPP4gvVKLoot5Y3uquLYpLg7F2CkT6Wekc

File Attributes: nftAttribute1 : undefined

Owner History: 0xf6ab0381201FE3f9727C52BE4D00812e62717385

NFT 조회하기

토큰 판매하기 1

NFT 판매하기

토큰 구매하기 2 1

NFT 구매하기

Token Id to search: 1 [main.js:498](#)
Token's owner address found: 0xf6ab0381201FE3f9727C52BE4D00812e62717385

Token name found: 자연
Token url found: QmRYRwedf9NwsVMin5k3N1MUESggCXBPCn64TrgNEjLayM

Token's attributes found: nftAttribute1 : undefined

Token found successfully [main.js:512](#)

Token Id to search: 0 [main.js:498](#)
Token's owner address found: 0xf6ab0381201FE3f9727C52BE4D00812e62717385

Token name found: je
Token url found: QmTbS2HvLnvjkPP4gvVKLoot5Y3uquLYpLg7F2CkT6Wekc

Token's attributes found: nftAttribute1 : undefined

Token found successfully [main.js:512](#)

>

Transaction: 0x3833bae4a063fa6f23ba58ac1b0c0b85894636b943e8d18e961a07398f3a7623
Gas usage: 63697
Block Number: 4
Block Time: Mon Jun 12 2023 22:09:37 GMT+0900 (대한민국 표준시)

토큰 판매하기

NFT 판매하기

Buyer Address: 0xf6ab0381201FE3f9727C52BE4D00812e62717385, [main.js:556](#)
Token ID to buy: 1, Payed value in wei: 200000000000000000

토큰 구매하기

NFT 구매하기

✖ GET https://ipfs.io/ipfs/QmRYRwedf9NwsVMin5k3N1MUESggCXBPCn64TrgNEjLayM:1 [🔗](#)
Rwedf9NwsVMin5k3N1MUESggCXBPCn64TrgNEjLayM 504
✖ GET https://ipfs.io/ipfs/QmTbS2HvLnvjkPP4gvVK...uLYpLg7F2CkT6Wekc:1 [🔗](#)
S2HvLnvjkPP4gvVKLoot5Y3uquLYpLg7F2CkT6Wekc 504

Transaction: 0x58099c852498ce55b872bf6cccfe3bcfa1c5458599586b17d91c69f79da19b46e
Gas usage: 23795
Block Number: 5
Block Time: Mon Jun 12 2023 22:09:57 GMT+0900 (대한민국 표준시)
Runtime Error: revert
Revert reason: Not enough ETH

Buyer Address: 0xf6ab0381201FE3f9727C52BE4D00812e62717385, [main.js:556](#)
Token ID to buy: 1, Payed value in wei: 500000000000000000

Pk inputted: [main.js:421](#)
0x365ed54817828fc3fdb00e41f049e56a74efe7f3c804767963028eefedc9b967

wallet import with private key [main.js:432](#)

Pk inputted: [main.js:421](#)
0x2cd9b78ad1c1ce005f57be5bd3bff980fbd3f4d4464ef22adb9b009f4c336c

wallet import with private key [main.js:432](#)

Buyer Address: 0xDd3fCb1C1E62fcE86f8FD1Ea8697608e640b7100, [main.js:556](#)
Token ID to buy: 1, Payed value in wei: 500000000000000000

eth_chainId
eth_getTransactionCount
eth_sendRawTransaction

Transaction: 0xea4c645493986d0fb611238515febd97f50b519f54a1df2c2e410721228e2a1c
Gas usage: 24800
Block Number: 6
Block Time: Mon Jun 12 2023 22:11:24 GMT+0900 (대한민국 표준시)
Runtime Error: revert
Revert reason: You cannot buy your own token

트랜잭션 판매 등록 & 충분하지 않은 토큰이라 거부 & 자신이 사려 하자 거부

지갑 주소: 0xDd3fCb1C1E62fcE86f8FD1Ea8697608e640b7100

보유 ETH: 100

보유 NFT: 0

토큰 구매하기

2

1

0x2cd9b78ad1c1ce005f57be

지갑 가져오기

NFT 구매하기

Pk inputted: [main.js:421](#)
0x365ed54817828fc3fdb00e41f049e56a74efe7f3c804767963028eefedc9b967

wallet import with private key [main.js:432](#)

Pk inputted: [main.js:421](#)
0x2cd9b78ad1c1ce005f57be5bd3bff980fbdaa3f4d4464ef22adb9b009f4c336c

wallet import with private key [main.js:432](#)

Buyer Address: 0xDd3fCb1C1E62fcE86f8FD1Ea8697608e640b7100, [main.js:556](#)
Token ID to buy: 1, Payed value in wei: 5000000000000000000

NFT buy succeeded [main.js:578](#)

>

Transaction: 0xfcaef5b0035cf78c65c6dec2d4d3a62d562298c2782c4915f9f171cf6c6c81a5
Gas usage: 80563
Block Number: 7
Block Time: Mon Jun 12 2023 22:12:20 GMT+0900 (대한민국 표준시)

[eth_getTransactionReceipt](#)

다른 지갑으로 NFT 구매

토큰 ID

1

Owner address 0xDd3fCb1C1E62fcE86f8FD1Ea8697608e640b7100

Name 자연

File path QmRYRwedf9NwsVMin5k3N1MUESggCXBpCn64TrgNEjLayM

File Attributes: nftAttribute1 : undefined

Owner History: 0xf6ab0381201FE3f9727C52BE4D00812e62717385 -> 0xDd3fCb1C1E62fcE86f8FD1Ea8697608e640b7100

NFT 조회하기

NFT 조회를 통해 실제로 이동이 있는지 확인