|  |  |
| --- | --- |
| 클레이튼 NFT 만들기 | |
| 작 성 일 자 | 2022.05.01 |
| 작 성 자 | 권가영 |
| 개 요 | NFT 만드는 과정을 설명한다. |

|  |
| --- |
| 1. **개념 정리**  * ERC721   + NFT의 표준으로 NFT라고 불리는 것들은 대부분 이 형식을 따른다.   + ERC-20 토큰과는 반대로 '대체 불가능'한 특징을 가지고 있다.   + NFT는 ERC20에 토큰 ID, 메타데이터 JSON 파일이 추가된 형태이다.   + 토큰 ID: NFT에 붙는 개별 식별 번호이다.   + 메타데이터 JSON: NFT에 넣을 정보를 담는 그릇 * KIP17   + KIP17은 위의 ERC721와 거의 동일   + 대체 불가능한 토큰 표준(NFT, Non-fungible token)은 고유한 자산을 나타내는 특수한 토큰 유형이다.   + 대 체 불가 토큰을 구현하기 위해 KIP-17을 사용한다 * IPFS   + 분산 저장소라는 의미로 위변조가 불가능하며 블록체인과 비슷하다.   + 이미지를 토큰화하여 (= NFT를 만든다.) 직접 블록체인에 저장하게 되면 매우 비싸진다.   + 따라서, NFT를 위한 데이터는 위변조가 되면 안되니까 블록체인과 비슷하고 안정적인 저장 시스템 IPFS가 나오게 되었다.  1. **클레이튼 NFT 만드는 과정**  * 필요한 오픈소스 다운로드  1. KIP17 오픈소스(소스코드)를 다운로드 받는다. 이는 위에 설명한 것과 같다.      1. 이미지파일(.png)를 준비한다. (boy.png)      1. metadata.json 파일을 준비한다. (아래의 내용을 담고 있다.)   {  "attributes": [  {  "trait\_type": "Breed",  "value": "boy"  },  {  "trait\_type": "color",  "value": "Mocha"  }  ],  "description": "This is a picture of a boy raising his hand.",  "image": " ipfs:// ",//여긴 CID 주소가 들어가는 부분이다. Pinata를 통해 받아올 예정  "name": "bpy"   * Pinata API(IPFS) 준비   + IPFS를 쉽게 쓸 수 있게 도와주는 프로그램이다.  1. Pinata 웹 사이트를 접속한다. (<https://www.pinata.cloud>)      1. Pinata에 회원가입을 한다. (비밀번호를 쳐도 유효하지 않다는 메시지가 떠서 추천 비밀번호를 사용해주었다.)      1. 이메일 인증을 한다. 그러면, 회원가입이 완료가 된다. (이메일으로 온 확인 코드를 넣어주었다.)      * IPFS(Pinata)에 이미지와 metadata.json 파일 올리기  1. Pinata에 로그인했다면 이미지 파일(boy.png)을 업로드한다. (업로드가 완료되었다면 CID 주소를 확인할 수 있다. (맨 마지막 사진을 확인)          1. CID주소를 복사해 Metadata.json 파일의 이미지 경로에 넣는다. (ipfs://를 추가 해야한다.)      * + metadata.json 파일의 내용은 아래와 같다.   {  "attributes": [  {  "trait\_type": "Breed",  "value": "boy"  },  {  "trait\_type": "color",  "value": "Mocha"  }  ],  "description": "This is a picture of a boy raising his hand.",  "image": "ipfs://QmVLSjCxYxY6n3LuGRz8Wj1bE62P4ZgQXf32ecHj5d6Dhq", //이미지의 CID를 넣어준다.  "name": "bpy"  }   1. 위에 작성한 metadata.json 파일 또한 ipfs에 업로드한다. (위의 이미지 파일을 업로드한 방식과 동일하다.)        * 소스코드를 블록체인에 올리기  1. Klaytn IDE에 접속한 후 전에 다운받아둔 KIP17을 복사하여 파일에 복사/붙여넣기한다. (3\_Ballot.sol 파일의 내용을 삭제하고 넣어주었다.)        1. 파일의 이름도 3\_Ballot.sol에서 3\_Nft.sol로 바꾸어주었다. (Rename)      1. KIP17의 맨 끝으로 와서 NFT 이름을 바꾸어준다. (KIP17TokenOwnable -> KIP17CoinNFT)   //바꾼 코드 내용이다.  contract KIP17CoinNFT is KIP17Full, KIP17MintableOwnable, KIP17MetadataMintableOwnable, KIP17Burnable, KIP17Pausable {      constructor (string memory name, string memory symbol) public KIP17Full(name, symbol) {      }  }   1. 컴파일을 해준다. Solidity의 버전인 ^0.5.0과 동일하기 컴파일러 버전도 맞춰준다.      1. EVM version을 설정한다. (london은 invalid evm version requested 에러 발생 -> byzantium으로 바꾸어주었음.)      1. 메타마스크에서 개인키를 받아온다. (비공개키 내보내기를 눌러준다.)      1. 받아온 개인키를 넣는다.      1. 위에서 설정한 NFT이름을 찾아 선택한다. (-> KIP17CoinNFT)      1. deploy하면 블록체인에 올라간다. (deploy하기 전에 deploy 설정 값에 NFT이름을 바꾸고 SYMBOL을 CRN으로 바꾸어주었다.)      1. **문제 발생 (정리)**  * insufficient funds of the sender for value 에러 발생   + 이 에러는 잔고 부족하거나 chain\_id가 틀릴 때 종종 발생함. * 현재 내 주소에서의 잔고가 0 Ether -> testnet으로 Ether를 받아오고자 함.        * 하지만 받아지지 않음. (결국 해결 못함.) |