

Blockchain technológiák és alkalmazások házi feladat

Non-Fungible Token

Tervezési döntések

Implementáltuk az ERC721 Non-Fungible Token Standard kompatibilis interfészt, úgy, hogy eleget tegyünk az összes feltételének, valamint megvalósítottuk a kapott feladat elvárásait is. Az alkalmazással többek között lehetséges új tokeneket létrehozni, megsemmisíteni, eladásra bocsátani, megvenni, másnak rendelkezési jogot adni a saját tokenjeink felett.

Az alkalmazás Solidity nyelven íródott, elkészítéséhez a [Truffle](#) keretrendszert használtuk.

Fontos megvalósítási részletek

NFT vásárlásakor többletfizetés esetén nincs visszajáró, a teljes összeget az eladó fél kapja.

Új NFT létrehozásakor az alapértelmezetten nem eladó. Akkor válik azzá, ha a tulajdonos beállít neki egy árat. Az NFT tulajdonosának megváltozása esetén ugyancsak nem lesz eladó.

Az ERC-721 interfész megvalósításához az alábbi dokumentumot vettük segítségül:
<https://github.com/0xcert/ethereum-erc721/blob/master/src/contracts/tokens/nf-token.sol>

A tesztesetek megvalósítása és futtatása

Az általunk készített NFT szerződés minden publikus metódusát teszteltük. Törekedtünk arra, hogy ne csak az egyszerűbb esetek tesztelve, ezért több összetett tesztet is megvalósítottunk. Összesen 27 teszteset készült.

A tesztek egy része Solidity, a másik JavaScript nyelven íródott.

A tesztesetek futtatása:

1. Truffle keretrendszer telepítéséhez ***npm install -g truffle*** parancs futtatása
2. ***truffle test*** futtatása a projekt gyökérkönyvtárában

Az okosszerződés API-ja

Minden kényes művelet ellenőrzi, hogy a hívó jogosult-e annak elvégzésére.

burn(uint256 _tokenId)

Megsemmisíti a kapott azonosítójú NFT-t.

mint(string memory _uri)

Létrehoz egy új NFT-t a kapott URI-val, amely tulajdonosa a hívó lesz. Minden új NFT alapértelmezetten nem eladó, ezen később lehet változtatni.

forSale(uint256 _tokenId, uint256 price)

Egy létező NFT-t eladásra kínál, amennyiben a hívó jogosult a kapott azonosítójú NFT módosítására. Az átadott ár a legkisebb ár, amivel meg lehet azt venni.

notForSale(uint256 _tokenId)

Eltávolítja az NFT árát, így azt nem lehet megvenni.

safeTransferFrom(address _from, address _to, uint256 _tokenId, _data)

Ingyenesen átruházza az NFT tulajdonjogát az átadott _to címnek. Ellenőrzi, hogy a _to cím fogadhat-e NFT-t.

safeTransferFrom(address _from, address _to, uint256 _tokenId)

Ingyenesen átruházza az NFT tulajdonjogát az átadott _to címnek. Ellenőrzi, hogy a _to cím fogadhat-e NFT-t. A _data értékét ""-re állítja.

transferFrom(address _from, address _to, uint256 _tokenId)

Ingyenesen átruházza az NFT tulajdonjogát az átadott _to címnek.

buy(uint256 _tokenId)

Amennyiben eladó az adott NFT, valamint minimum az árának megfelelő mennyiségű Ether kerül küldésre a hívó által, az NFT átkerül a vevő címére.

approve(address _approved, uint256 _tokenId)

Az adott NFT módosítási jogát adja a kapott cím.

setApprovalForAll(address _operator, bool _approved)

Az _operator címnek jogosultságot adhat vagy vehet el a hívó az ő minden tulajdonában álló NFT-jére.

getPrice(uint256 _tokenId)

Visszaadja, hogy a kapott azonosítójú NFT mennyibe kerül, amennyiben az eladó.

balanceOf(address _owner)

Visszaadja, hogy a kapott cím hány NFT-vel rendelkezik.

ownerOf(uint256 _tokenId)

Visszaadja a címet, amely birtokolja a kapott azonosítóval rendelkező NFT-t.

getApproved(uint256 _tokenId)

Visszaadja, hogy a kapott azonosítójú NFT-nek melyik cím a meghatalmazottja, aki a tulajdonoson kívül még módosíthatja azt.

isApprovedForAll(address _owner, address _operator)

Visszaadja, hogy az _operator címnek van-e jogosultsága az _owner cím összes NFT-jének módosítására.