

# Non-Fungible Token megvalósítása Solidity nyelven

A megoldásunk, illetve a pontos feladatkiírás megtekinthető az alábbi linken: <https://github.com/KostyalBalint/Nft-Solidity>

## Környezet

*Visual Studio* fejlesztőkörnyezetében, a telepített *Solidity* pluginnal dolgoztunk a munkafolyamat során.

A projekt futtatásához és teszteléséhez a Github-on található pontosított leírás.

## Működési elv

A token egy okos szerződésből áll, melynek alapját az *Openzeppelin* könyvtár adja. Így elegendő csak a számunkra fontos függvényeket implementálni. Ilyen módon használjuk az **ERC-721** szabvány funkcióit.

## Token létrehozása - minting

A token-t bárki a hálózaton tudja mint-elni megfelelő mennyiségű (0.01) Ether befizetésével. Ilyenkor szükséges megadni egy URL-t ami az Nft-hez tartozó képre mutat. Jelen megoldás szerint nem lehet két egyforma url-t mint-elni.

Egy-egy tokneről le lehet kérdezni a hozzá tartozó URL-t, és mivel ez csak egy lekérdező függvény, így Gas-t sem kell fizetni a futtatásáért.

## Token mozgatása - transfer

A token tulajdonosa bármikor átadhatja a token birtoklásának jogát egy másik felhasználónak. Ehhez csupán a tranzakciós díjat kell megfizetni. Ez a metódus egy ajándékozásnak felel meg és a kapó félnek semmilyen fizetési kötelezettsége nincsen.

## Token adásvétele - buy and sale

Az eladni kívánt token a tulajdonosa a `setForSale()` függvény segítségével tudja megvásárolhatóvá tenni, az ár megadásával. A `removeFromSale()` használatával pedig visszavonhatja eladási szándékát.

Adott tokenhez minden esetben tartozik eladási ár. Amennyiben ez az ár 0, az egyenértékű azzal, hogy a token nem valójában nem eladó.

Ebből következik, hogy ingyenes token nem létezik.

Adott token árát a `getTokenPrice()` segítségével lehet megtudni

A token vásárolni kívánó felhasználó az előzőekben az eladó által megjelölt összegért megveheti a token, melyhez rendelkezésére áll a `buy()` funkció. Ebben az esetben az eladó félnek utal a kép árával megegyező mennyiségű ethert, majd ő lesz a vagyontárgy birtokosa.

## Tesztelés Truffle segítségével

A Token fő funkcióinak tesztelését Truffle segítségével végezzük.

A tesztelés és fejlesztés a Ganache applikáció által biztosított teszhálózaton történt.

A tesztek futtathatók az `npm test` paranccsal.

### Futtatott tesztek:

```
Contract: ImageNft
deployment
  ✓ should deploy successfully
  ✓ should have a name and symbol (308ms)
minting
  ✓ should cost 0.01 ether to mint a token (325ms)
  ✓ should revert a transaction with lower ether (1933ms)
  ✓ should have proper uri string of a minted token (662ms)
  ✓ should throw an error for minting the same token (513ms)
transfer
  ✓ should be transferable between users (752ms)
  ✓ should revert when transfer is not sent from the owner (616ms)
buy and sell
  ✓ should be sellable (443ms)
  ✓ should be the token set for sale for the given price (528ms)
  ✓ should allow only the owner to set a token for sale (562ms)
  ✓ should be able to buy token (305ms)
  ✓ should cost the current amount of money to buy a token (378ms)
  ✓ should not be able to buy a token which is not for sale (609ms)
  ✓ should remove the token from sale (1012ms)
  ✓ should allow only the owner to remove the token from sale (802ms)
```