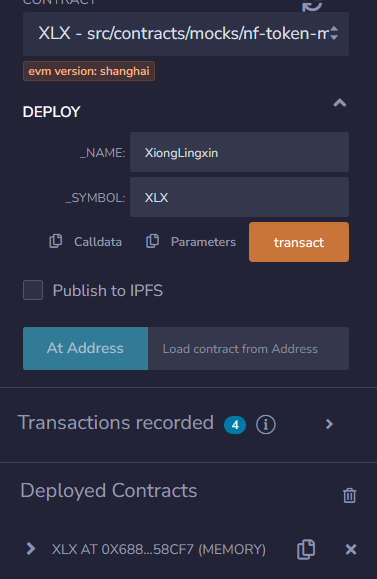
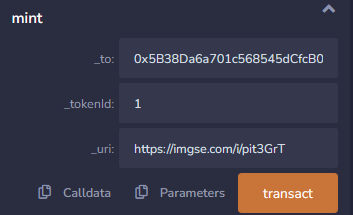
熊灵欣/2021131126/区块链工程214

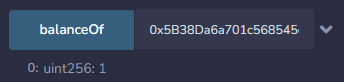
1. 准备三个地址：
   1. 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
   2. 0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2
   3. 0x4B20993Bc481177ec7E8f571ceCaE8A9e22C02db
2. 图床图片地址：<https://imgse.com/i/pit3GrT>
3. 部署合约名字为XLX



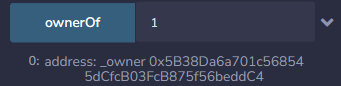
1. mint一个tokenid为1到账户A



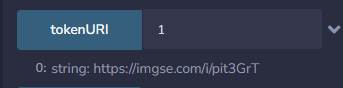
1. 查看账户A余额



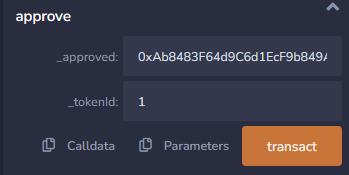
1. 查看tokenid为1的owner



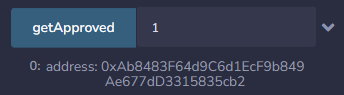
1. 查看tokenid为1的tokenURI



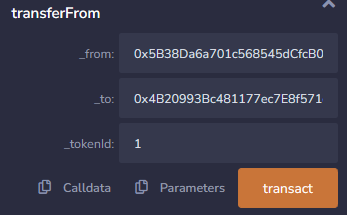
1. 使用账户A对账户B进行approve



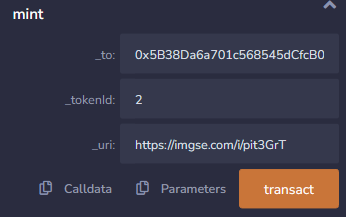
1. 调用getApproved，查看tokenid为1的返回值



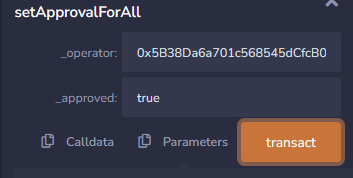
1. 使用账户B调用transferFrom，把tokenid为1的NFT从账户A转到账户C；



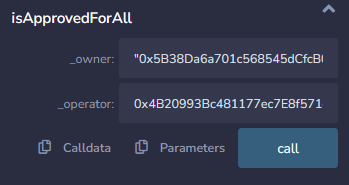
1. mint tokenid为2 的nft到账户A



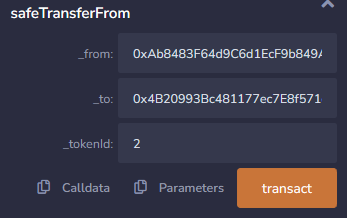
1. 使用账户A调用setApprovalForAll，对账户B进行approve



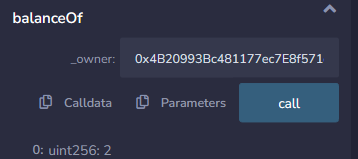
1. 调用isApprovedForAll，查看授权状态



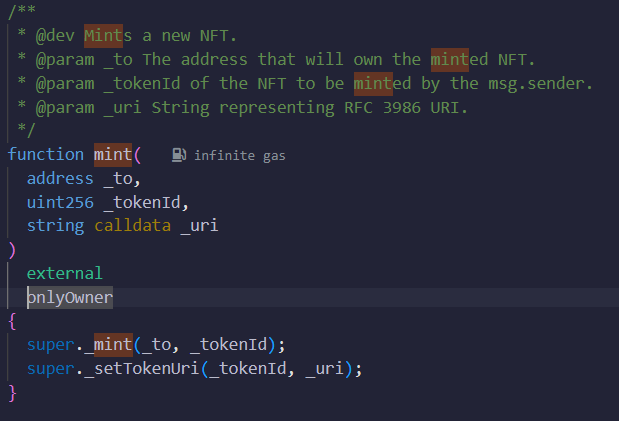
1. 使用账户B，转移tokenid为2的NFT到账户C；



1. 查看账户C的余额



1. Mint已经是OnlyOwner



1. ERC20的transfer事件中的参数是：

event Transfer(address indexed \_from, address indexed \_to, uint256 \_value)；

ERC721的transfer事件中的参数是：

event Transfer(address indexed \_from, address indexed \_to, uint256 indexed \_tokenId);

异：ERC20中参数是value（转账数量），ERC721中是tokenId（唯一标识符）

同：ERC20 和 ERC721 需要发送方、接收方的地址。

这是因为：

①在ERC20代币中，每个代币是可互换的，没有区别，因此转移事件只需要记录发送方、接收方和转移数量即可。代币的价值主要体现在数量上，而不是特定的代币实例。

②在ERC721代币中，每个代币都是独特的，具有唯一的标识符。因此，在转移事件中需要记录发送方、接收方以及转移的代币的唯一标识符（Token ID）。这是因为ERC721代币通常代表不同的资产或收藏品。

1. ERC20的approval事件中的参数是：

event Approval(address indexed \_owner, address indexed \_spender, uint256 \_value)

ERC721的approval事件中的参数是：

event Approval(address indexed \_owner, address indexed \_approved, uint256 indexed \_tokenId);

异：ERC20中参数是value（转账数量），ERC721中是tokenId（唯一标识符）

同：ERC20 和 ERC721 都需要代币的所有者地址、被授权的地址。

这是因为：

①在ERC20代币中，授权通常用于允许其他地址从授权者的余额中转移一定数量的代币。因此，授权事件需要记录被授权的地址和被授权的数量，以便其他地址可以在授权限额范围内进行转移。

②在ERC721代币中，授权通常用于允许其他地址对代币进行操作，如交易、转移或更改代币的属性。由于ERC721代币的每个代币都是独特的，没有数量的概念，因此授权事件只需要记录被授权的地址即可，无需记录数量。