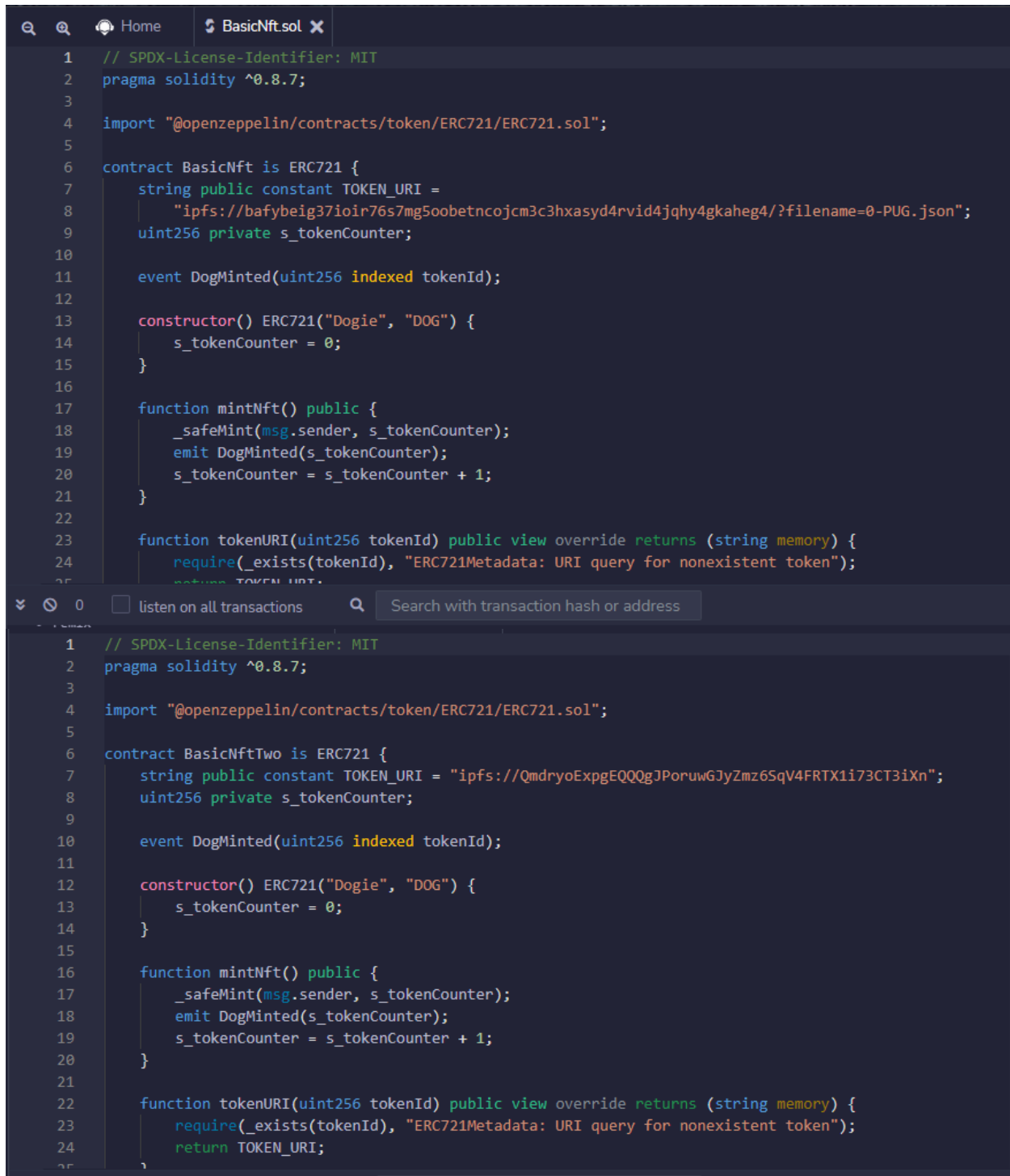


NextJS NFT Marketplace



```
1 // SPDX-License-Identifier: MIT
2 pragma solidity ^0.8.7;
3
4 import "@openzeppelin/contracts/token/ERC721/ERC721.sol";
5
6 contract BasicNft is ERC721 {
7     string public constant TOKEN_URI =
8         "ipfs://bafybeig37ioir76s7mg5oobetncojcm3c3hxasyd4rvid4jqhy4gkaheg4/?filename=0-PUG.json";
9     uint256 private s_tokenCounter;
10
11     event DogMinted(uint256 indexed tokenId);
12
13     constructor() ERC721("Dogie", "DOG") {
14         s_tokenCounter = 0;
15     }
16
17     function mintNft() public {
18         _safeMint(msg.sender, s_tokenCounter);
19         emit DogMinted(s_tokenCounter);
20         s_tokenCounter = s_tokenCounter + 1;
21     }
22
23     function tokenURI(uint256 tokenId) public view override returns (string memory) {
24         require(_exists(tokenId), "ERC721Metadata: URI query for nonexistent token");
25         return TOKEN_URI;
26     }
27 }
```

0 ☐ listen on all transactions Search with transaction hash or address

```
1 // SPDX-License-Identifier: MIT
2 pragma solidity ^0.8.7;
3
4 import "@openzeppelin/contracts/token/ERC721/ERC721.sol";
5
6 contract BasicNftTwo is ERC721 {
7     string public constant TOKEN_URI = "ipfs://QmdryoExpqEQQqGJPoruwGJyZmz6SqV4FRTX1i73CT3iXn";
8     uint256 private s_tokenCounter;
9
10     event DogMinted(uint256 indexed tokenId);
11
12     constructor() ERC721("Dogie", "DOG") {
13         s_tokenCounter = 0;
14     }
15
16     function mintNft() public {
17         _safeMint(msg.sender, s_tokenCounter);
18         emit DogMinted(s_tokenCounter);
19         s_tokenCounter = s_tokenCounter + 1;
20     }
21
22     function tokenURI(uint256 tokenId) public view override returns (string memory) {
23         require(_exists(tokenId), "ERC721Metadata: URI query for nonexistent token");
24         return TOKEN_URI;
25     }
26 }
```

```
1 // SPDX-License-Identifier: MIT
2 pragma solidity ^0.8.7;
3
4 import "@openzeppelin/contracts/token/ERC721/IERC721.sol";
5 import "@openzeppelin/contracts/security/ReentrancyGuard.sol";
6
7 // Check out https://github.com/Fantom-foundation/Artion-Contracts/blob/5c90d2bc0401af6fb5abf35b860b76
8 // For a full decentralized nft marketplace
9
10 error PriceNotMet(address nftAddress, uint256 tokenId, uint256 price);
11 error ItemNotForSale(address nftAddress, uint256 tokenId);
12 error NotListed(address nftAddress, uint256 tokenId);
13 error AlreadyListed(address nftAddress, uint256 tokenId);
14 error NoProceeds();
15 error NotOwner();
16 error NotApprovedForMarketplace();
17 error PriceMustBeAboveZero();
18
19 contract NftMarketplace is ReentrancyGuard {
20     struct Listing {
21         uint256 price;
22         address seller;
23     }
24
25     event ItemListed(
26         address indexed seller,
27         address indexed nftAddress,
28         uint256 indexed tokenId,
29         uint256 price
30     );
31
32     event ItemCanceled(
33         address indexed seller,
34         address indexed nftAddress,
35         uint256 indexed tokenId
36     );
37
38     event ItemBought(
39         address indexed buyer,
40         address indexed nftAddress,
41         uint256 indexed tokenId,
42         uint256 price
43     );
44
45     mapping(address => mapping(uint256 => Listing)) private s_listings;
46     mapping(address => uint256) private s_proceeds;
```