

# Forum:[Hack Forums](#)

Thread: [Create Your Own NFT - ERC721 Solidity Smart Contract](#)

Hackers - May 16, 2024, 5:45 am

## Fallen ERC721 NFT Smart Contract Example

Today you will learn how to create an ERC-721 smart contract. This is meant for Ethereum/EVM-based blockchain tokens, using Solidity as the core language. This is a basic example to help you get started, but does work as-is. You can deploy the contract and as the owner, you can mint NFTs to other addresses. The contract below (source code in spoiler at end of post) does not include IPFS or other onchain data, but instead acts as a mechanism to distribute tokens. You can use these tokens however you wish in external systems, such as using them for access, memberships, real-world representations, or other use cases. Feel free to add on features and extend this base contract however you wish to implement custom functions!

Code // SPDX-License-Identifier: MIT

pragma solidity ^0.8.25;First we can define the license for it (MIT) and then the Solidity version number. In this case I am using 0.8.25, but I have tested this with 0.8.0+ and it works there too.

Code import "@openzeppelin/contracts/token/ERC721/ERC721.sol";

import "@openzeppelin/contracts/access/Ownable.sol";We use the ERC721 contract from openzeppelin so we don't have to recreate the wheel for NFT ownership, transfers, approvals, minting, security, and other features.

We use Ownable to enforce on-chain security for token ownership so people can't manage other people's tokens and to keep things secure.

Code contract FallenExampleERC721 is ERC721, Ownable {Your contract name can be whatever you want, but in this case it is Fallen Example ERC721 - I always prefer camelCase for this but with the first letter capitalized too.

Code uint256 private \_tokenIdCounter;We can keep track of how many tokens have been minted like this - keep in mind that if someone burns their NFT, a token ID may not be valid anymore, but the total will always increment based on the minting - it should not be used as a totalcount of active NFTs because of burning. The newest token ID will always be +1 from the last, starting at 1 in this contract. If you wanted you could have it start at a different number, but this breaks best practice for what people expect.

Code constructor() ERC721("FallenExampleERC721", "FE721") {}The constructor is run when the contract is first deployed. In this case we are just setting it as an ERC721 contract, to mint/manage our custom NFT. We do nothing else inside of the {} code block.

Code function safeMint(address to) public onlyOwner {  
 uint256 tokenId = \_tokenIdCounter;  
 \_tokenIdCounter += 1;  
 \_safeMint(to, tokenId);

}Finally, our safeMint function acts as the public/external func for people to use to mint a new NFT. You can see onlyOwner attribute/flag on the function, which means only the owner of the smart contract can use it. You can use this to mint new NFTs to any other address. Remember, this is a super basic example to help you get started - you can create a contract that has an onchain fee/price for each token if you want people to buy & mint, or you can even do public minting with no extra price. You can pre-mint all tokens to yourself if you want to transfer instead of minting in future, or you could come up with other unique rules for how these NFTs are minted.

We set the token ID to the current token ID count, which is the newest #, starting from 1 on first mint, then 2, 3, and so on.

We then increment the token id counter before we mint , so that the count is already +1 for the next mint that occurs on the contract. This is a safeguard / best practice to update data related to mints before the mint. The same is true if you are deducting balances - deduct from contract balance before sending. This is due to a thing called reentrancy but I won't go deep into that here. Just know that you should follow this rule 100% of the time.

The \_safeMint function corresponds to an openzeppelin method to mint the NFT to the destination address, for the new token ID.

Full source code:

Spoiler ([Click to View](#))

Code // SPDX-License-Identifier: MIT

pragma solidity ^0.8.25;

```
import "@openzeppelin/contracts/token/ERC721/ERC721.sol";
```

```
import "@openzeppelin/contracts/access/Ownable.sol";
```

```
contract FallenExampleERC721 is ERC721, Ownable {  
    uint256 private _tokenIdCounter;
```

```
    constructor() ERC721("FallenExampleERC721", "FE721") {}
```

```
    function safeMint(address to) public onlyOwner {
```

```
        uint256 tokenId = _tokenIdCounter;
```

```
        _tokenIdCounter += 1;
```

```
        _safeMint(to, tokenId);
```

```
    }
```

```
}
```

Looking for more? Check this out:

<https://hackforums.net/showthread.php?tid=6271197>

V?rt - May 16, 2024, 6:27 am

What an informative thread. Great job explaining everything.

Hackers - May 16, 2024, 7:28 am

(May 16, 2024 06:27 AM)V?rt Wrote: [\[CITE\]](#) What an informative thread. Great job explaining everything.

Thanks, hope this helps someone. Many people don't realize how easy it is to get started w/ this

stuff. The contracts for the most basic versions of NFTs or other token types can be written with a very small amount of code.

**LeetSpeed** - May 16, 2024, 9:58 am

Thanks for this! This is valuable information!

**Hackers** - May 16, 2024, 4:41 pm

(May 16, 2024 09:58 AM)LeetSpeed Wrote: [\[CITE\]](#) Thanks for this! This is valuable information!

You are welcome!

Have you written much Solidity in the past?

**idalistrades** - May 16, 2024, 4:50 pm

Thank you

I plan to use this for my NFT game in the future

**7** - May 16, 2024, 5:51 pm

Another mind blowing thread nodes!

Great information, looking forward to understanding this by practise

**LeetSpeed** - May 16, 2024, 9:27 pm

(May 16, 2024 04:41 PM)nodes Wrote: [\[CITE\]](#) You are welcome!

Have you written much Solidity in the past?

No not too much.

**Hackers** - May 17, 2024, 4:45 am

(May 16, 2024 04:50 PM)FemaleAI\_CEO Wrote: [\[CITE\]](#) Thank you

I plan to use this for my NFT game in the future

awesome! You should look into ERC 1155 too because you can do batch minting which may be better and it works very similarly.

(May 16, 2024 05:51 PM)NulldB Wrote: [\[CITE\]](#) Another mind blowing thread nodes!

Great information, looking forward to understanding this by practise

Thank you!

If you want to chat about anything just let me know

(May 16, 2024 09:27 PM)LeetSpeed Wrote: [\[CITE\]](#) No not too much.

Ah, not too hard to get into, but it does help if you have prior programming experience of some sort. A cool language though and easier to bug test / fix than others in my opinion because of how strict Solidity is with types & logic.

**crocsss** - May 20, 2024, 11:11 am

excuse me can i send you a pm on discord?

**LeetSpeed** - May 22, 2024, 9:32 am

(May 17, 2024 04:45 AM)nodes Wrote: [\[CITE\]](#) awesome! You should look into ERC 1155 too because you can do batch minting which may be better and it works very similarly.

Thank you!

If you want to chat about anything just let me know

Ah, not too hard to get into, but it does help if you have prior programming experience of some sort.

A cool language though and easier to bug test / fix than others in my opinion because of how strict Solidity is with types & logic.

We'll be in contact I have a few questions for you.

**Hackers** - May 22, 2024, 9:48 am

(May 20, 2024 11:11 AM)crocsss Wrote: [\[CITE\]](#) excuse me can i send you a pm on discord? please reply in this thread or message onsite.

(May 22, 2024 09:32 AM)LeetSpeed Wrote: [\[CITE\]](#) We'll be in contact I have a few questions for you.

no problem, feel free to reachout via convo or reply here!

**Absence** - May 22, 2024, 9:48 am

Thank you for another high quality contribution [Hackers](#)

**Hackers** - May 24, 2024, 3:47 am

(May 22, 2024 09:48 AM)Absence Wrote: [\[CITE\]](#) Thank you for another high quality contribution [Hackers](#)

thanks Absence, let me know if you want help w/ anything.

I don't do as much web3/blockchain work in the last few months but figure I should practice with a couple contracts every couple of months so I don't lose knowledge about this stuff. I'm not a huge fan of the average NFT, but the technology itself can be very useful in real world for digitalized asset ownership & more. I'm sure as AI takes off we'll see more and more crypto-based microservices and tokenized access passes.