Solana Dev 101 - How to Mint an NFT on Solana

Introduction

GM! In this blog we are going to dig into the world of Non-Fungible Tokens (NFTs) on Solana. By the end of this post you should have a clear understanding of the process and be able to create your own NFTs on Solana using JavaScript.

If you have any questions, don't hesitate to join the

[Helius Discord

](https://discord.com/invite/HjummjUXgq) or tag [Helius on Twitter

](https://twitter.com/heliuslabs) for assistance.

Prerequisites:

Before we start, make sure you have Node.js and npm installed on your machine. You can download them [here

](https://nodejs.org/en/download/).

Step 1: Setup

First, we need to install the necessary dependencies. Run the following commands in your terminal to install the Solana JavaScript API and Metaplex JavaScript library:

Step 2: Importing Dependencies

Next, we can create a new file called mintNFTs.js and we can start off by importing the required dependencies:

Step 3: Create a Wallet , Initialise a Connection, and Create a Metaplex Instance

We need to create a Solana wallet which will be used to mint and hold the NFT. To achieve this we will use the Solana CLI tools, you can run the following command in the terminal to get a new wallet. This wallet will be saved as a JSON file containing the seed phrase.

You can import this into a browser wallet extension such as Phantom or Backpack.

⚠WARNING ⚠ : Make sure not to share your private key with anyone.

Now, we initialize a connection to the Solana cluster. We will be using Helius to connect to a mainnet cluster as this will be faster than the public RPC which is highly rate limited and does not have archival capabilities. (https://api.mainnet-beta.solana.com). If you would also like to use a faster and more premium RPC service instead of the standard RPC you can create a free Helius account here.

We will then create a wallet key pair from the seed phrase of the wallet that we just created.

In the above code we create an instance of the Metaplex SDK, a toolkit for interacting with the Metaplex protocol. We then configure the instance with a connection to a Solana cluster, the user's wallet, and a storage provider. .use(bundlrStorage())

configures the instance to use Bundlr as the storage provider, which is where the metadata and assets of the NFTs will be stored.

Step 4: Create and Upload the NFT Metadata

In this step we delve into the creation and uploading of the NFT metadata and image through the createMetadata()

function.

One of the core parts of an NFT is the image, in this example we will be using this image of the Helius Logo! Make sure to add the image that you want to use to the directory where you have created the mintNFT.js file.

This function starts by reading the image file specified by imageName

usingfs

, then converts it into a format suitable for Metaplex with toMetaplexFile()

. We then upload the metadata using the metaplex.nfts().uploadMetadata()

function. The metadata contains important details about our NFT, including its name , a description , the image we prepared, and a set of attributes that provide additional details about the NFT. There is a lot more data you can include in your NFT, you can read more about the Metaplex NFT standard [here

](https://docs.metaplex.com/programs/token-metadata/token-standard#the-non-fungible-standard).

After the metadata is successfully uploaded the function will return the URI of the metadata which points to the online location of the data.

Step 5: Minting the NFT

Now we move forward with minting the NFT by creating a function called createNFT()

. Initially, we invoke the createMetadata()

function with "./heliusLogo.png" as the argument (path to the image file). Next, we mint the NFT by calling metaplex.nfts().create()

. This function takes in an object with several properties such as the metadata URI, the name of the NFT, a seller fee (set here at 5%), and an array of creators, where each creator object includes the address and a share percentage of the amount of royalties that wallet should get. After successfully creating the NFT, the function will log the mint address of the NFT. This mint address is a unique identifier for the NFT on the blockchain. You can copy this address and look up the NFT on a block explorer such as XRAY.

Running this code with "node mintNFT.js

" created the NFT that can be seen [here

](https://xray.helius.xyz/token/7BwvG5tjbCJ7vvCes5tAEovsurvP1Mk8sTbotnJn2d6v).

Conclusion

Congrats! You have just created an NFT. I hope this tutorial was helpful in getting you started with NFTs on Solana. NFTs are a fundamental aspect of the Solana ecosystem so understanding how they are structured and how to create them is a great skill to have.

You can find the complete code [here

](https://github.com/owenventer/SolanaDev101/blob/main/mintAnNFT/mintNFT.js). Happy coding!

References:

- [Metaplex JS SDK documentation

](https://github.com/metaplex-foundation/js/blob/main/README.md)

- [Metaplex token standard

](https://docs.metaplex.com/programs/token-metadata/token-standard#the-non-fungible-standard)