Chapter 4

Tokens

Introduction

The Solana SPL Token program is a powerful tool provided by the Solana program library that allows developers to create and manage tokens on the Solana blockchain. Tokens, in general, represent digital assets or units of value within a blockchain ecosystem. They enable a wide range of applications such as digital currencies, rewards systems, in-game assets, and more.

SPL tokens are a specific implementation of tokens on the Solana blockchain. They adhere to a set of standards defined by the SPL token program. These standards ensure proper token functionality and compatibility within the Solana ecosystem. By conforming to these standards, SPL tokens become interoperable with other applications and wallets built on Solana.

SPL tokens support both fungible and non-fungible token types, giving developers the flexibility to create various types of digital assets on the Solana blockchain. With the capability to represent both uniform and unique assets, SPL tokens offer exciting opportunities for developers to build innovative decentralized applications spanning from financial systems to decentralized marketplaces.

SPL vs. ERC-20 Tokens

The ERC-20 token contract is the standard for fungible tokens on Ethereum, and other EVM-based chains. While Solana's SPL Token program is, like ERC-20, a smart contract, that's about where the similarities end. They both create functions to manage tokens and their supply, but SPL tokens are fundamentally different. The root of the difference lies in Solana's account-based model, which Ethereum does not have. So, for an ERC-20 token, ownership data is merely assigned to an EOA (a wallet) by means of a lookup table (a database) that accompanies the token's contract. Conversely, for an SPL token, an account is created with the token's details (mint ID, decimals, and token balance) inside. The token account is assigned an "owner" which signs for and controls the account. *Note: an SPL token account must be created for a wallet before tokens can be sent to it. This requires a small amount of rent to open, and is recoverable after the token account is closed.*

Fungible Tokens

One of the key aspects of SPL tokens is their ability to be either fungible or non-fungible. Fungible tokens are identical and interchangeable with each other. They have the same value and properties, allowing for easy exchange between holders. Examples of fungible tokens include cryptocurrencies like Bitcoin and Ether. These tokens are commonly used for transactions, trading, and as a medium of exchange. As an exercise, we will show how to create a token with 1B supply using the command line.

First, we'll need some SOL. We'll use the testnet cluster to test this out.

```
solana config set --url testnet
# Config File: /home/username/.config/solana/cli/config.yml
# RPC URL: https://api.testnet.solana.com
# WebSocket URL: wss://api.testnet.solana.com/ (computed)
# Keypair Path: ./my-keypair.json
# Commitment: confirmed

solana airdrop 1
# Requesting airdrop of 1 SOL
# Signature:
5updStDRqLWpDgcu2HoSZQmx4DGCESkthcJm8bQpxHzE85gWL2r5GAF8i6JpaY12R14BqPRstesYA4cr
RvuG9Wv8
# 1 SOL
```

To mint a new SPL token using the Solana CLI spl-token utility with a supply of 1,000,000,000, we'll follow the following steps:

1. First, make sure you have the Solana CLI installed and connected to a Solana cluster.
2. Create a new token account by running the following command:

```
spl-token create-token
# Creating token 4Wnsn6wnjZhBz22bfBQQkvu39zT24hUXExxUDA5mAowb
#
# Address:  4Wnsn6wnjZhBz22bfBQQkvu39zT24hUXExxUDA5mAowb
# Decimals:  9
#
# Signature:
4zdWNGnorYCcwWupjE7F3woWNfaMmUj8VWqWpqLp41njCVz24cPbp3Yjacoc15Cpsc2eRKSTbUGSP2Y4
Fshy5LpS
```

3. Next, we need to create the token account to hold newly minted tokens:

```
spl-token create-account 4Wnsn6wnjZhBz22bfBQQkvu39zT24hUXExxUDA5mAowb
# Creating account Ah6PC9WaHmEG8zyhNpVovnmQ5qqURkozVFVRqP1vbhFa
#
# Signature:
39CPVsaKc1UTAKs9N4381MsEGiPDHNzZxzcMgNUU7vxQMmRfXUaU3UqMFjN7K4kBY6zxHEDNVyvFCAHD
91JXstaG
```

This command will generate a new token account address for the token you are going to mint.

3. Mint the new token with the desired supply using the following command:

```
spl-token mint <token-account-address> <supply> --authority
<your-wallet-address>
# in our case:
spl-token mint 4Wnsn6wnjZhBz22bfBQQkvu39zT24hUXExxUDA5mAowb 1000000000
Ah6PC9WaHmEG8zyhNpVovnmQ5qqURkozVFVRqP1vbhFa
# Minting 1000000000 tokens
# Token: 4Wnsn6wnjZhBz22bfBQQkvu39zT24hUXExxUDA5mAowb
# Recipient: Ah6PC9WaHmEG8zyhNpVovnmQ5qqURkozVFVRqP1vbhFa
#
# Signature:
5ZSknZKBpzYkLLekJCSUmmQxMrTTX5Z2nJFpvC9p67L7Xri9q92FrZQSvchNtowk3fpJFgs58K7qZ4aA
3bDo658v
```

Replace with the address of the token account you created in step 2. should be set to 1000000000 to mint 1,000,000,000 tokens should be replaced with the Solana wallet address that you want to grant authority to mint these tokens.

4. Verify the token supply and balance by running the following command:

```
spl-token supply <token-account-address>
spl-token balance <token-account-address>
```

These commands will display the total supply of the token and the balance of the token account, respectively.

Non-Fungible Tokens (NFTs)

A modification to the fungible tokens [explained above] are non-fungible tokens (NFTs). NFTs are unique and indivisible tokens (one-of-one) that represent ownership of a specific asset or piece of data. Unlike fungible tokens, NFTs have distinct characteristics and cannot be exchanged on a one-to-one basis. Each NFT holds unique data and serves as a proof of authenticity and ownership for digital assets such as artwork, collectibles, or really anything. In the previous chapter (Wallets), we described NFTs, as well as several unique flavors of NFT including executable (xNFT) and compressed (cNFT). The technology for NFT creation, distribution, and collection management is evolving incredibly quickly, and is squarely outside of this author's competence. Our explanation would not do this space justice, so suffice it to say the ecosystem is vibrant, and we encourage you to venture out and see for yourself. We will update this section of the book soon.

Token2022

To expand creator expressivity and the capabilities of SPL tokens, the Token2022 standard has been implemented. The standard has an enormous breadth of new controls and features, we will try to enumerate some of them here. The release marks the introduction of various extensions in Token2022, such as `Transfer Hook`, `MintCloseAuthority`, `InterestBearingMint`, `NonTransferable mint`, and `Confidential Transfers`, which enhance the functionality and privacy of SPL tokens. The standard introduces the use of homomorphic encryption, specifically "Twisted ElGamal Encryption," to enable hidden mathematical operations on ciphertext, ensuring the confidentiality of token balances and transfer amounts. Finally, the standard brings the implementation of Sigma Protocols, a class of zero-knowledge proofs, to validate the correctness and privacy of different instructions and proofs related to Confidential Transfers, such as public-key validity, ciphertext validity, zero-balance proof, equality proof, fee sigma proof, and the range proof. We will leave the explanation at that, and refer the reader to consult this article https://www.helius.dev/blog/what-is-token-2022 created by the kind folks at Helius Labs.

Recap

The Solana SPL Token program is a crucial component of the Solana blockchain that allows developers to create and manage tokens. SPL tokens conform to specific standards, enabling compatibility and interoperability within the Solana ecosystem. These tokens can be either fungible, representing identical and interchangeable assets, or non-fungible, representing unique and indivisible assets. Advanced token types like the Backpack team's *xNFT*, or *Token22*-enhanced SPL tokens create greater expressivity and control for team's launching their tokens on Solana. The ability to support any of these tokens empowers developers to build diverse applications on Solana, ranging from simple currencies to dynamic digital assets that will power Web3.