Send and Receive Tokens with the Solana CLI

This page describes how to receive and send SOL tokens using the command line tools with a command line wallet such as apaper wallet , afile system wallet , or ahardware wallet . Before you begin, make sure you have created a wallet and have access to its address (pubkey) and the signing keypair. Check out ourconventions for entering keypairs for different wallet types .

Testing your Wallet

Before sharing your public key with others, you may want to first ensure the key is valid and that you indeed hold the corresponding private key.

In this example, we will create a second wallet in addition to your first wallet, and then transfer some tokens to it. This will confirm that you can send and receive tokens on your wallet type of choice.

This test example uses our Developer Testnet, called devnet. Tokens issued on devnet haveno value, so don't worry if you lose them.

Airdrop some tokens to get started

First,airdrop yourself some play tokens on the devnet.

solana airdrop 1

< RECIPIENT_ACCOUNT_ADDRESS

--url https://api.devnet.solana.com where you replace the textwith your base58-encoded public key/wallet address.

A response with the signature of the transaction will be returned. If the balance of the address does not change by the expected amount, run the following command for more information on what potentially went wrong:

solana confirm -v < TRANSACTION_SIGNATURE

Check your balance

Confirm the airdrop was successful by checking the account's balance. It should output1 SOL :

solana balance < ACCOUNT_ADDRESS

--url https://api.devnet.solana.com

Create a second wallet address

We will need a new address to receive our tokens. Create a second keypair and record its pubkey:

solana-keygen new --no-passphrase --no-outfile The output will contain the address after the textpubkey: . Copy the address. We will use it in the next step.

pubkey: GKvqsuNcnwWqPzzuhLmGi4rzzh55FhJtGizkhHaEJqiV You can also create a second (or more) wallet of any type:paper ,file system , orhardware .

Transfer tokens from your first wallet to the second address

Next, prove that you own the airdropped tokens by transferring them. The Solana cluster will only accept the transfer if you sign the transaction with the private keypair corresponding to the sender's public key in the transaction.

solana transfer --from < KEYPAIR

< RECIPIENT_ACCOUNT_ADDRESS

0.5 --allow-unfunded-recipient --url https://api.devnet.solana.com --fee-payer < KEYPAIR

where you replace with the path to a keypair in your first wallet, and replacewith the address of your second wallet.

Confirm the updated balances withsolana balance :

solana balance < ACCOUNT_ADDRESS

--url http://api.devnet.solana.com where is either the public key from your keypair or the recipient's public key.

Full example of test transfer

solana-keygen new --outfile my_solana_wallet.json

Creating my first wallet, a file system wallet

Generating a new keypair For added security, enter a passphrase (empty for no passphrase) : Wrote new keypair to my_solana_wallet.json == pubkey: DYw8jCTfwHNRJhhmFcbXvVDTqWMEVFBX6ZKUmG5CNSKK

Here is the address of the first wallet

==

==

==

==

==

==

==

==

==

==

==

==

==

==

==

==

==

==

== Save this seed phrase to recover your new keypair: width enhance concert vacant ketchup eternal spy craft spy guard tag punch

If this was a real wallet, never share these words on the internet like this!

==

==

==

==

==

==

==

==

==

==

==

==

==

==

==

==

==

==

==

==

solana airdrop 1 DYw8jCTfwHNRJhhmFcbXvVDTqWMEVFBX6ZKUmG5CNSKK --url https://api.devnet.solana.com

Airdropping 1 SOL to my wallet's address/pubkey

Requesting airdrop of 1 SOL from 35.233 .193.70:9900 1 SOL

solana balance DYw8jCTfwHNRJhhmFcbXvVDTqWMEVFBX6ZKUmG5CNSKK --url https://api.devnet.solana.com

Check the address's balance

1 SOL

solana-keygen new --no-outfile

Creating a second wallet, a paper wallet

Generating a new keypair For added security, enter a passphrase (empty for no passphrase) : == pubkey: 7S3P4HxJpyyigGzodYwHtCxZyUQe9JiBMHyRWXArAaKv

Here is the address of the second, paper, wallet.

==

==

==

==

==

==

==

==

==

==

==

==

==

==

==

==

==

Save this seed phrase to recover your new keypair: clump panic cousin hurt coast charge engage fall eager urge win love

If this was a real wallet, never share these words on the internet like this!

==

==

==

==

==

==

==

==

==

==

==

==

==

==

==

==

==

==

solana transfer --from my_solana_wallet.json 7S3P4HxJpyyigGzodYwHtCxZyUQe9JiBMHyRWXArAaKv 0.5 --allow-unfunded-recipient --url https://api.devnet.solana.com --fee-payer my_solana_wallet.json

Transferring tokens to the public address of the paper wallet

3gmXvykAd1nCQQ7MjosaHLf69Xyaqyq1qw2eu1mgPyYXd5G4v1rihhg1CiRw35b9fHzcftGKKEu4mbUeXY2pEX2z

This is the transaction signature

solana balance DYw8jCTfwHNRJhhmFcbXvVDTqWMEVFBX6ZKUmG5CNSKK --url https://api.devnet.solana.com 0.499995 SOL

The sending account has slightly less than 0.5 SOL remaining due to the 0.000005 SOL transaction fee payment

solana balance 7S3P4HxJpyyigGzodYwHtCxZyUQe9JiBMHyRWXArAaKv --url https://api.devnet.solana.com 0.5 SOL

The second wallet has now received the 0.5 SOL transfer from the first wallet

Receive Tokens

To receive tokens, you will need an address for others to send tokens to. In Solana, the wallet address is the public key of a keypair. There are a variety of techniques for generating keypairs. The method you choose will depend on how you choose to store keypairs. Keypairs are stored in wallets. Before receiving tokens, you will need tocreate a wallet . Once completed, you should have a public key for each keypair you generated. The public key is a long string of base58 characters. Its length varies from 32 to 44 characters.

Send Tokens

If you already hold SOL and want to send tokens to someone, you will need a path to your keypair, their base58-encoded public key, and a number of tokens to transfer. Once you have that collected, you can transfer tokens with thesolana transfer command:

solana transfer --from < KEYPAIR

< RECIPIENT_ACCOUNT_ADDRESS

< AMOUNT

--fee-payer < KEYPAIR

Confirm the updated balances withsolana balance :

solana balance < ACCOUNT_ADDRESS

Previous Solana CLI: Test Validator Next Solana Validator Architecture