Associated Token Account Program

This program defines the convention and provides the mechanism for mapping the user's wallet address to the associated token accounts they hold.

Motivation

- A user may own arbitrarily many token accounts belonging to the same mint
- which makes it difficult for other users to know which account they should send
- tokens to and introduces friction into many other aspects of token management.
- This program introduces a way todeterministically
- derive a token account key
- from a user's main System account address and a token mint address, allowing the
- user to create a main token account for each token they own. We call these
- accountsAssociated Token Accounts
- .
- In addition, it allows a user to send tokens to another user even if the
- beneficiary does not yet have a token account for that mint. Unlike a system
- transfer, for a token transfer to succeed the recipient must have a token
- account with the compatible mint already, and somebody needs to fund that token
- account. If the recipient must fund it first, it makes things like airdrop
- campaigns difficult and just generally increases the friction of token
- transfers. The Associated Token Account program allows the sender to create the associated token account for
- the receiver, so the token transfer just works.

See theSPL Token program for more information about tokens in general.

Background

Solana's programming model and the definitions of the Solana terms used in this document are available at:

- https://docs.solana.com/apps
- https://docs.solana.com/terminology

Source

The Associated Token Account Program's source is available onGitHub .

Interface

The Associated Token Account Program is written in Rust and available oncrates.io anddocs.rs .

Finding the Associated Token Account address

The associated token account for a given wallet address is simply a program-derived account consisting of the wallet address itself and the token mint.

Theget_associated_token_address Rust function may be used by clients to derive the wallet's associated token address.

The associated account address can be derived in TypeScript with:

import

{ PublicKey }

from

'@solana/web3.js' ; import

{

TOKEN_PROGRAM_ID

}

from

'@solana/spl-token' ;

const

SPL_ASSOCIATED_TOKEN_ACCOUNT_PROGRAM_ID : PublicKey =

new

PublicKey ('ATokenGPvbdGVxr1b2hvZbsiqW5xWH25efTNsLJA8knL' ,) ;

function

findAssociatedTokenAddress (walletAddress : PublicKey , tokenMintAddress : PublicKey) : PublicKey { return PublicKey . findProgramAddressSync ([walletAddress . toBuffer ( ) , TOKEN_PROGRAM_ID . toBuffer ( ) , tokenMintAddress . toBuffer ( ) ,] , SPL_ASSOCIATED_TOKEN_ACCOUNT_PROGRAM_ID) [0] ; }

Creating an Associated Token Account

If the associated token account for a given wallet address does not yet exist, it may be created byanybody by issuing a transaction containing the instruction returned bycreate_associated_token_account .

Regardless of creator the new associated token account will be fully owned by the wallet, as if the wallet itself had created it.
Edit this page Previous Token-Lending Program Next Token Upgrade Program * Motivation * Background * Source * Interface * * Finding the Associated Token Account address* * Creating an Associated Token Account