Token Factory Tutorial

Thetokenfactory module enables any account to create new tokens with a unique identifier based on the creator's address. Each account can generate multiple tokens by specifying distinct denoms. The token creator is granted "admin" privileges, allowing them to mint, burn, and transfer their token.

These tokens are namedfactory/{CREATOR_ADDRESS}/{DENOM} and come with a range of native functionalities.

Requirements

To create a token on the devnet, ensure you have the following setup:

- Theseid
- CLI
- A wallet with SEI tokens on devnet

You can obtain devnet tokens from one of the faucets listedhere .

Creating a Denom

seid

tx

tokenfactory

create-denom DENOM --from=ACCOUNT

--chain-id=arctic-1

--node=https://rpc.arctic-1.seinetwork.io/

--broadcast-mode=block

--fees=20000usei This command creates a new coin with the formatfactory/{ACCOUNT}/{DENOM} . ReplaceDENOM with your desired denom name andACCOUNT with your account name. The account specified here will be the token admin.

Simplify your tutorial experience by setting theDENOM andACCOUNT variables in your terminal. This allows you to easily refer to these variables throughout the tutorial.

For instance, to set a denom calledsolid and use the account name of your choice, use the following commands:

DENOM

solid ACCOUNT = your_account_name Replaceyour_account_name with the actual name of your account. Once set, you can refer to these values throughout the session usingDENOM andACCOUNT .

Understanding Command Line Arguments

When executing commands in this tutorial, you'll encounter several arguments. Here's a brief overview of what each means:

- --chain-id=arctic-1
- : This specifies the network where the command will be executed. In this case,arctic-1
- is the identifier for the Sei devnet.
- --node=https://rpc.arctic-1.seinetwork.io/
- : This points to the RPC URL of the node you are interacting with.
- --broadcast-mode=block
- : This determines how your transaction is broadcasted to the network. Theblock
- mode means the transaction will wait to be included in a block before returning a response. This is a safer option as it confirms your transaction is processed.
- --fees=20000usei
- : This is used to specify the transaction fee.

Understanding these arguments will help you execute the commands more confidently and customize them as needed for different scenarios.

For detailed descriptions of these arguments, useseid help in the CLI.

Minting Tokens

seid

tx

tokenfactory

mint AMOUNT --from=ACCOUNT

--chain-id=arctic-1

--node=https://rpc.arctic-1.seinetwork.io/

--broadcast-mode=block

--fees=20000usei This command will create (mint) a specificAMOUNT of your new token. ReplaceAMOUNT with the number of tokens you want to mint, followed by the token denom generated from the previous command.

For instance, if you would like to mint 1M tokens, you should input1000000factory/{ACCOUNT}/{DENOM} as amount or you could set a new variable for the amount:

AMOUNT

1000000 factory/ {ACCOUNT} / {DENOM} To verify that the tokens have been minted, query the balance of your account:

seid

query

bank

balances ACCOUNT --chain-id=arctic-1

--node=https://rpc.arctic-1.seinetwork.io/

Burning Tokens

seid

tx

tokenfactory

burn AMOUNT --from=ACCOUNT

--chain-id=arctic-1

--node=https://rpc.arctic-1.seinetwork.io/

--broadcast-mode=block

--fees=20000usei This command allows you to burn a specific amount of your tokens, reducing the total supply. To use it, replaceAMOUNT with the number of tokens you wish to destroy.

For instance, to burn 100 tokens, you should input100factory/{ACCOUNT}/{DENOM} . Ensure that you substitute{ACCOUNT} and{DENOM} with your actual account address and token denomination, respectively. You can also update the variable:

AMOUNT

100 factory/ {ACCOUNT} / {DENOM} Only the token admin has permission to mint and burn tokens. If necessary, you can reassign these privileges by using thechange-admin command to designate a new admin.

Create Pointer Contract

To enable seamless use of this token in EVM environments, we can create a pointer contract. This process results in an ERC20 token that can be imported and used in EVM wallets and applications.

seid

tx

evm

deploy-erc20 DENOM NAME SYMBOL DECIMAL --from=ACCOUNT

--evm-rpc=https://evm-rpc.arctic-1.seinetwork.io/ Parameters

- DENOM
- : The denomination of the token for which you want to create an ERC20 pointer. This should match the TokenFactory token you created.
- DENOM
- : The name you wish to assign to your ERC20 pointer token. It should match the name of the TokenFactory token.
- SYMBOL
- : The symbol for your ERC20 pointer token. It should correspond with the symbol of the TokenFactory token.
- DECIMAL
- : The number of decimals for your ERC20 pointer token. This should align with the decimals of the TokenFactory token (typically 6).

Flags

- --from
- : The Sei address from which the deployment transaction is sent. This address must have enough balance to cover transaction fees.
- --evm-rpc
- : The endpoint URL for the EVM RPC interface of the Sei blockchain. This URL is used by theseid
- command to interact with the Sei EVM.

Executing this command creates an ERC20 token and outputs the contract address. This token is linked to the TokenFactory token, meaning any activities involving this ERC20 token will also reflect on the state of the TokenFactory token and vice versa.

Learn more about EVM interoperability and pointer contractshere .

Next Steps

Congrats on completing the Token Factory tutorial! You've learned how to create, mint, and burn tokens on Sei using thetokenfactory module.

Smart contracts can also create TokenFactory denoms and act as token admins, allowing for more complex and automated token management strategies. For more advanced features and detailed insights, please refer to theToken Factory module documentation(opens in a new tab).

If you encounter a bug while using the devnet, please submit it via the formhere . Last updated onMay 24, 2024EVM (CLI) NFT Contracts