

Token Allowlists

The [Sei Token Factory module](#) enables token creators to restrict who can interact with their custom tokens through the use of allowlists.

This tutorial will guide you through understanding the configuration and limitations of allowlists, creating a new denom with an allowlist, updating an existing denom to add or modify an allowlist, and querying the allowlist of a denom.

Prerequisites

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

- Theseid
- CLI installed.
- A wallet with SEI tokens on devnet.
- Access to a running Sei blockchain node.

You can obtain devnet tokens from one of the faucets listed [here](#). Command Line Argument Overview When executing commands in this tutorial, you'll encounter several arguments. Here's a brief overview of what each means:

- `--allow-list ALLOW_LIST_FILE_PATH`
 - : This specifies the path to the allowlist file that we have created in the previous step. E.g. `allowlist.json`
 - .
- `--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. The `block` 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, use `theseid help` in the CLI.

Validation Rules & Limits

- * **Valid Addresses** : All addresses in the allowlist must be valid in their respective formats (SEI Bech32 or EVM 0x prefixed). Invalid addresses will result in a validation error.
- * **Address Conversion** : EVM addresses (0x prefixed) are automatically converted to SEI Bech32 addresses when the allowlist is processed.
- * **No Duplicates** : Duplicate addresses in the allowlist will be removed automatically.
- * **Maximum Size** : By default, the maximum number of addresses allowed in the allowlist is 2,000. This limit can be configured by the chain administrator by setting `tokenfactory.denom_allow_list_max_size` in node config.
- * **Empty List** : If you wish to remove the allowlist, you need to pass an empty list object `{ "addresses": [] }`.
- * **Additional Considerations** : Ensure to handle error cases and validation properly when interacting with the allowlist.

Creating a Denom with Allowlist

1. First, create a JSON file that contains the list of addresses you want to include in your allowlist.
2. The format of the allowlist is as follows:
3.

```
{
```
4.

```
  "addresses"
```
5.

```
  :
```
6.

```
  [
```
7.

```
    "address1"
```
8.

```
    ,
```
9.

```
    "address2"
```
10.

```
    ,
```
11.

```
    "address3"
```
12.

```
  ]
```
13.

```
}
```
14. Ensure that all addresses are valid SEI (Bech32) or EVM (0x-prefixed) addresses. Note that 0x addresses will be converted to Bech32 addresses when the allowlist is persisted.
15. The maximum number of addresses allowed in the allowlist is 2000 by default. This can be configured by the chain administrator.
16. Create a new denom with the specified allowlist.

```

17. seid
18. tx
19. tokenfactory
20. create-denom
21. SUBDENOM \
22. --allow-list=ALLOW_LIST_FILE_PATH \
23. --from=CREATOR_ACCOUNT \
24. --chain-id=CHAIN_ID \
25. --fees=FEE_AMOUNT \
26. --gas=GAS_LIMIT \
27. -y
28. Here is an example
29. seid
30. tx
31. tokenfactory
32. create-denom
33. mytoken
34. \
35. --allow-list=./allow_list.json \
36. --from
37. mykey
38. \
39. --chain-id
40. sei-chain
41. \
42. --fees
43. 10000
44. usei
45. \
46. --gas
47. auto
48. \
49. -y

```

Updating an Existing Denom Allowlist

To update the allowlist of a token, you can use the following command:

```

seid

tx

tokenfactory

update-denom SUBDENOM \ --allow-list=ALLOW_LIST_FILE_PATH \ --from=ACCOUNT \ --chain-id=CHAIN_ID \ --
node=NODE_RPC_URL \ --broadcast-mode=block \ --fees=FEE_AMOUNT \ --gas=GAS_LIMIT \ -y Here is an example
seid

tx

tokenfactory

update-denom

mytoken \ --allow-list=./updated_allow_list.json \ --from

mykey \ --chain-id

arctic-1 \ --node

https://rpc.arctic-1.seinetwork.io/ \ --broadcast-mode=block \ --fees

20000 usei \ --gas

auto \ -y To re-enable all addresses to transfer the token, you can simply submit an empty allowlist.

```

Querying a Denom Allowlist

You may query the allowlist of a token using the node REST endpoint. Here is an example:

```
curl
```

```
-X
```

```
'GET' \ 'https://rest-arctic-1.sei-apis.com/sei-protocol/seichain/tokenfactory/denoms/allow_list?  
denom=factory/{ACCOUNT}/{DENOM}' \ -H
```

```
'accept: application/json'
```

EVM Support

To enable usage of this token in EVM environments, you can create a pointer contract. The detailed process is outlined in the [Token Factory tutorial](#).

For more advanced features and in-depth information, refer to the [Token Factory module documentation \(opens in a new tab\)](#).

Last updated on October 11, 2024 [Token Factory NFT Contracts](#)