

Contents

Token CLI Commands Reference	3
Global Options	3
Top-Level Commands	3
Ping	3
ApiVersion	3
NetworkInfo	3
Util Commands	3
util create-account	4
util calculate-address	4
util check-address	4
util encode-wallet	4
util decode-wallet	4
util digest	5
util sign-digest	5
util sign-message	5
util recover-address	5
util balance-of	5
util send	6
Task Commands	6
task info	6
task list	6
task approve	6
task execute	6
task collect	7
task abort	7
Token Commands	7
token deploy	7
token migrate	8
token address	8
token code-version	8
token settings	9
token supply	9
token balance-of	9
token send	9
token mint	9
token burn	10
token freeze	10
token unfreeze	10
token is-frozen	11
token enable-transfer	11
token disable-transfer	11
token is-transfer-enabled	11
token set-symbol	11
token set-decimals	12

Tokens Commands	12
tokens list	12
tokens code-version	12
tokens has-code-version	12
tokens list-codes	12
tokens upload-code	13
Transaction Commands	13
transaction status	13
transaction wait	13
Fee Commands	13
fee settings	13
fee is-fee-enabled	14
fee get-vip-rate	14
fee get-default-rate	14
fee get-account-rate	14
fee calculate-fee	14
fee is-vip-account	14
fee get-min-fee	14
fee get-max-fee	15
fee get-current-total-fee-in-pool	15
fee get-total-collected-fee	15
fee set-fee-settings	15
fee enable	16
fee disable	16
fee set-account-rate	16
fee set-vip-account	16
fee withdraw	17
Permission Commands	17
permission settings	17
permission get-default-permission	17
permission get-token-permission	17
permission is-account-locked	17
permission get-mask	18
permission get-kyc-level	18
permission can-transfer	18
permission create-permission	18
permission set-permission	18
permission enable-permission	19
permission disable-permission	19
permission set-mode	19
permission lock-account	20
permission unlock-account	20
permission set-mask	20
permission set-kyc-level	21
Role Commands	21
role is-root	21

role get-root	21
role is-role	22
role get-role-members	22
role change-root	22
role set-role	22
role unset-role	23
Common Task Options	23
Role Types	23
KYC Levels	23
Notes	24

Token CLI Commands Reference

An CLI utility for ethereum token management.

Global Options

- `--api-url <URL>` - Token API URL to connect to (default: `http://localhost:4000`, env: `TOKEN_API_URL`)
- `--state-manager-url <URL>` - State Manager URL to connect to (default: `http://localhost:3030`, env: `STATE_MANAGER_URL`)

Top-Level Commands

Ping

Ping the Token API server.

```
token-cli ping
```

ApiVersion

Display the Token API version.

```
token-cli api-version
```

NetworkInfo

Display network information.

```
token-cli network-info
```

Util Commands

Utility sub-commands for account and wallet management.

Global Option: - `--remote` - Use remote API to perform sensitive operations (default: false)

util create-account

Create a new Ethereum account.

```
token-cli util [--remote] create-account [SECRET]
```

Arguments: - SECRET - Optional secret to derive the account from

Note: Accounts created locally will not match those created by the remote API due to different derivation methods.

util calculate-address

Calculate Ethereum address from private key.

```
token-cli util [--remote] calculate-address <PRIVATE_KEY>
```

Arguments: - PRIVATE_KEY - Private key in hex format (with or without 0x prefix)

util check-address

Check if an Ethereum address is valid.

```
token-cli util check-address <ADDRESS>
```

Arguments: - ADDRESS - Ethereum address to check

util encode-wallet

Encode a JSON Keystore wallet from private key and password.

```
token-cli util [--remote] encode-wallet --private-key <PRIVATE_KEY> --password <PASSWORD> [-
```

Options: - `--private-key <PRIVATE_KEY>` - Private key in hex format - `--password <PASSWORD>` - Password to encrypt the wallet - `--dir <DIR>` - Output directory to save the keystore file (default: current directory) - `--filename <FILENAME>` - Output file name to save the keystore JSON (default: auto-generated)

util decode-wallet

Decode a JSON Keystore wallet to retrieve the private key.

```
token-cli util [--remote] decode-wallet --file <FILE> --password <PASSWORD>
```

Options: - `--file <FILE>` - Input keystore file containing the keystore JSON - `--password <PASSWORD>` - Password to decrypt the wallet

util digest

Compute the Ethereum Signed Message Digest for a given input string.

```
token-cli util digest <MESSAGE>
```

Arguments: - MESSAGE - Input string to compute the digest for

util sign-digest

Sign an Ethereum Signed Message Digest with a private key.

```
token-cli util [--remote] sign-digest <DIGEST> --private-key <PRIVATE_KEY>
```

Arguments: - DIGEST - 32-byte hex string digest (with or without 0x prefix)

Options: - --private-key <PRIVATE_KEY> - Private key in hex format

Note: Signatures generated locally may vary from remote API due to different recovery ID handling.

util sign-message

Sign a message with a private key.

```
token-cli util [--remote] sign-message <MESSAGE> --private-key <PRIVATE_KEY>
```

Arguments: - MESSAGE - Input string message to sign

Options: - --private-key <PRIVATE_KEY> - Private key in hex format

util recover-address

Recover Ethereum address from signed message and signature.

```
token-cli util [--remote] recover-address <MESSAGE> <SIGNATURE>
```

Arguments: - MESSAGE - Message that was signed - SIGNATURE - Signature in hex format

util balance-of

Get the ether balance of an Ethereum address.

```
token-cli util balance-of <ACCOUNT> [--pretty] [--in-wei]
```

Arguments: - ACCOUNT - Ethereum address to check balance

Options: - --pretty - Pretty print the balance with commas (default: false) - --in-wei - Display the balance in wei instead of ether (default: false)

util send

Send ether from one account to another.

```
token-cli util send --from <FROM> --private-key <PRIVATE_KEY> --to <TO> --amount <AMOUNT> [
```

Options: - --from <FROM> - Ethereum address to send from - --private-key <PRIVATE_KEY> - Private key of the sender - --to <TO> - Ethereum address to send to - --amount <AMOUNT> - Amount to send (in ether by default, or wei if -in-wei is set) - --in-wei - Indicate if the amount is specified in wei instead of ether (default: false)

Task Commands

Commands for managing tasks.

task info

Get task information by its key.

```
token-cli task info <TASK_KEY>
```

Arguments: - TASK_KEY - Task key to get information for

task list

List tasks.

```
token-cli task list [--role <ROLE>] [--all]
```

Options: - --role <ROLE> - Filter tasks by role - --all - Show all tasks, including inactive ones (default: false)

task approve

Approve a task.

```
token-cli task approve <TASK_KEY> [--private-key <PRIVATE_KEY> | --wallet <WALLET> --password
```

Arguments: - TASK_KEY - Key of the task to approve

Options: - --private-key <PRIVATE_KEY> - Private key to sign the approval - --wallet <WALLET> - Input keystore wallet file - --password <PASSWORD> - Password for the keystore wallet (will prompt if not provided)

task execute

Execute a task.

```
token-cli task execute <TASK_KEY>
```

Arguments: - TASK_KEY - Key of the task to execute

task collect

Force a task into collectable state.

```
token-cli task collect <TASK_KEY>
```

Arguments: - TASK_KEY - Key of the task to collect

task abort

Abort a task by creating an abort task.

```
token-cli task abort <KEY_TO_ABORT> [--description <DESCRIPTION>] [--min-approvals <MIN_APPROVALS>]
```

Arguments: - KEY_TO_ABORT - Key of the task to abort

Options: - --description <DESCRIPTION> - Description for the abort task -
--min-approvals <MIN_APPROVALS> - Minimum number of approvals required
(default: 1)

Token Commands

Commands for managing a single token. All commands require --name <TOKEN_NAME>.

token deploy

Deploy a new Ethereum token (DEFAULT_ADMIN_ROLE only).

```
token-cli token --name <NAME> deploy --symbol <SYMBOL> [OPTIONS]
```

Required Options: - --symbol <SYMBOL> - Symbol of the token (e.g., MTK)

Token Configuration: - --decimals <DECIMALS> - Decimal places of the token
(default: 18) - --max-supply <MAX_SUPPLY> - Maximum supply (0 for unlimited)
- --initial-supply <INITIAL_SUPPLY> - Initial supply - --initial-holder
<ADDRESS> - Address of the initial token holder

Token Features: - --no-transfer - Disable token transfers (default: false) -
--frozen - Freeze the token after creation (default: false) - --with-permission
- Enable permission for the token (default: false) - --private-permission -
Enable private permission for the token (default: false) - --token-permission
<ADDRESS> - Address of token permission contract (default: ZERO_ADDRESS)
- --with-mask - Turn on permission mask feature (default: false) - --with-kyc
- Enable KYC feature (default: false) - --with-token-fee - Enable token fee
feature (default: false)

Fee Configuration: - --vip-rate <RATE> - VIP rate in basis points -
--default-rate <RATE> - Default rate in basis points - --min-fee <AMOUNT>

- Minimum fee for token transfers - `--max-fee <AMOUNT>` - Maximum fee for token transfers

Role Management: - `--isolated-roles` - Enable local account manager for role management (default: false) - `--account-man <ADDRESS>` - Private account manager address (default: ZERO_ADDRESS) - `--root <ADDRESS>` - Local root address for local account manager - `--default-admin-roles <ROLES>` - Comma-separated list of default admin roles - `--upgrader-admin-roles <ROLES>` - Comma-separated list of upgrader admin roles - `--mint-admin-roles <ROLES>` - Comma-separated list of minter admin roles - `--burn-admin-roles <ROLES>` - Comma-separated list of burner admin roles - `--compliance-admin-roles <ROLES>` - Comma-separated list of compliance admin roles - `--fee-admin-roles <ROLES>` - Comma-separated list of fee admin roles - `--fee-recipient-roles <ROLES>` - Comma-separated list of fee recipient roles - `--pauser-admin-roles <ROLES>` - Comma-separated list of pauser admin roles

Task Options: - `--min-approvals <COUNT>` - Minimum number of approvals required (default: 1) - `--auto-execute` - Automatically execute after reaching minimum approvals - `--expiry <SECONDS>` - Expiry time in seconds (default: 604800, 0 for no expiry) - `--collect-delay <SECONDS>` - Delay before task collection starts (default: 86400) - `--description <DESCRIPTION>` - Description for the deploy task

token migrate

Migrate an existing token to a new code version (UPGRADER_ROLE only).

```
token-cli token --name <NAME> migrate --new-version <VERSION> [OPTIONS]
```

Required Options: - `--new-version <VERSION>` - New token code version to migrate to

Task Options: - `--description <DESCRIPTION>` - Description for the migrate task - `--min-approvals <COUNT>` - Minimum number of approvals required (default: 1) - `--auto-execute` - Automatically execute after reaching minimum approvals - `--expiry <SECONDS>` - Expiry time in seconds (default: 604800) - `--collect-delay <SECONDS>` - Delay before task collection starts (default: 86400)

token address

Get the token contract address by token name.

```
token-cli token --name <NAME> address
```

token code-version

Get the code version of a deployed token.

```
token-cli token --name <NAME> code-version
```

token settings

Get the token settings by token name.

```
token-cli token --name <NAME> settings [--pretty] [--in-base-unit]
```

Options: - --pretty - Output raw balance in pretty format (default: false) -
--in-base-unit - Show balance as base units (true) or full tokens (false, default
- more user-friendly)

token supply

Get the current supply information of a token.

```
token-cli token --name <NAME> supply [--pretty] [--in-base-unit]
```

Options: - --pretty - Output raw balance in pretty format (default: false) -
--in-base-unit - Show balance as base units (true) or full tokens (false, default
- more user-friendly)

token balance-of

Get the balance of an account for the token.

```
token-cli token --name <NAME> balance-of <ADDRESS> [--pretty] [--in-base-unit]
```

Arguments: - ADDRESS - Account address to query balance for

Options: - --pretty - Output raw balance in pretty format (default: false) -
--in-base-unit - Show balance as base units (true) or full tokens (false, default
- more user-friendly)

token send

Send tokens from one account to another.

```
token-cli token --name <NAME> send --from <FROM> --to <TO> --amount <AMOUNT> [--in-base-unit]
```

Options: - --from <FROM> - Sender address - --to <TO> - Recipient address -
--amount <AMOUNT> - Amount to send (as string, in base units) -
--in-base-unit - Treat amount as base units (true) or full tokens (false, default
- more user-friendly) - --private-key <PRIVATE_KEY> - Private key of
the sender address

token mint

Mint new tokens to a specified account (MINT_ADMIN_ROLE only).

```
token-cli token --name <NAME> mint --to <TO> --amount <AMOUNT> [--in-base-unit] [OPTIONS]
```

Required Options: - --to <TO> - Recipient address - --amount <AMOUNT> -
Amount to mint (as string, e.g. '100' for 100 tokens or '1000000000000000000000'
if using -in-base-unit)

Task Options: - `--description <DESCRIPTION>` - Description for the unfreeze token task - `--min-approvals <COUNT>` - Minimum number of approvals required (default: 1) - `--auto-execute` - Automatically execute after reaching minimum approvals - `--expiry <SECONDS>` - Expiry time in seconds (default: 604800) - `--collect-delay <SECONDS>` - Delay before task collection starts (default: 86400)

token is-frozen

Check if the token is frozen.

```
token-cli token --name <NAME> is-frozen
```

token enable-transfer

Enable token transfers (PAUSER_ROLE only).

```
token-cli token --name <NAME> enable-transfer [OPTIONS]
```

Task Options: - `--description <DESCRIPTION>` - Description for the enable transfer task - `--min-approvals <COUNT>` - Minimum number of approvals required (default: 1) - `--auto-execute` - Automatically execute after reaching minimum approvals - `--expiry <SECONDS>` - Expiry time in seconds (default: 604800) - `--collect-delay <SECONDS>` - Delay before task collection starts (default: 86400)

token disable-transfer

Disable token transfers (PAUSER_ROLE only).

```
token-cli token --name <NAME> disable-transfer [OPTIONS]
```

Task Options: - `--description <DESCRIPTION>` - Description for the disable transfer task - `--min-approvals <COUNT>` - Minimum number of approvals required (default: 1) - `--auto-execute` - Automatically execute after reaching minimum approvals - `--expiry <SECONDS>` - Expiry time in seconds (default: 604800) - `--collect-delay <SECONDS>` - Delay before task collection starts (default: 86400)

token is-transfer-enabled

Check if token transfers are enabled.

```
token-cli token --name <NAME> is-transfer-enabled
```

token set-symbol

Set a new symbol for the token (UPGRADER_ROLE only).

```
token-cli token --name <NAME> set-symbol <SYMBOL> [OPTIONS]
```

Arguments: - SYMBOL - New symbol for the token

Task Options: - --min-approvals <COUNT> - Minimum number of approvals required (default: 1) - --auto-execute - Automatically execute after reaching minimum approvals - --expiry <SECONDS> - Expiry time in seconds (default: 604800) - --collect-delay <SECONDS> - Delay before task collection starts (default: 86400)

token set-decimals

Set token decimals (UPGRADER_ROLE only).

```
token-cli token --name <NAME> set-decimals <DECIMALS> [OPTIONS]
```

Arguments: - DECIMALS - New decimals for the token

Task Options: - --min-approvals <COUNT> - Minimum number of approvals required (default: 1) - --auto-execute - Automatically execute after reaching minimum approvals - --expiry <SECONDS> - Expiry time in seconds (default: 604800) - --collect-delay <SECONDS> - Delay before task collection starts (default: 86400)

Tokens Commands

Token factory management sub-commands.

tokens list

List all tokens.

```
token-cli tokens list
```

tokens code-version

Get the default code version of token factory.

```
token-cli tokens code-version
```

tokens has-code-version

Check if a token code version is uploaded.

```
token-cli tokens has-code-version <CODE_VERSION>
```

Arguments: - CODE_VERSION - Code version of the token contract to check

tokens list-codes

List all uploaded token contract code versions.

```
token-cli tokens list-codes
```

tokens upload-code

Create a task to upload token contract bytecode to the blockchain.

```
token-cli tokens upload-code --file-path <FILE_PATH> --code-version <CODE_VERSION> [OPTIONS]
```

Required Options: - --file-path <FILE_PATH> - File path of the token contract code - --code-version <CODE_VERSION> - Code version of the token contract being uploaded

Task Options: - --min-approvals <COUNT> - Minimum number of approvals required (default: 1) - --auto-execute - Automatically execute after reaching minimum approvals - --expiry <SECONDS> - Expiry time in seconds (default: 604800) - --collect-delay <SECONDS> - Delay before task collection starts (default: 86400) - --description <DESCRIPTION> - Description for the task

Transaction Commands

Transaction related sub-commands.

transaction status

Show the status of a transaction by its hash.

```
token-cli transaction status --txid <TXID>
```

Options: - --txid <TXID> - Transaction hash

transaction wait

Wait for a transaction to be confirmed.

```
token-cli transaction wait --txid <TXID> [--confirmations <CONFIRMATIONS>] [--poll-interval
```

Options: - --txid <TXID> - Transaction hash - --confirmations <CONFIRMATIONS> - Number of confirmations to wait for (default: 1) - --poll-interval <POLL_INTERVAL> - Polling interval in seconds (default: 10)

Fee Commands

Commands for managing token fee. All commands require --name <TOKEN_NAME>.

fee settings

Get token fee settings.

```
token-cli fee --name <NAME> settings [--pretty] [--in-base-unit]
```

Options: - `--pretty` - Pretty print amount (default: false) - `--in-base-unit`
- Show amount in base units

fee is-fee-enabled

Check if the token fee feature is enabled.

```
token-cli fee --name <NAME> is-fee-enabled
```

fee get-vip-rate

Get token fee rate for all VIPs (in basis points).

```
token-cli fee --name <NAME> get-vip-rate
```

fee get-default-rate

Get default token fee rate for all non-VIPs (in basis points).

```
token-cli fee --name <NAME> get-default-rate
```

fee get-account-rate

Get token fee rate for a specific account.

```
token-cli fee --name <NAME> get-account-rate --account <ACCOUNT>
```

Options: - `--account <ACCOUNT>` - Account address

fee calculate-fee

Calculate token fee for a transfer.

```
token-cli fee --name <NAME> calculate-fee --from <FROM> --to <TO> --amount <AMOUNT> [--pretty]
```

Options: - `--from <FROM>` - Sender address - `--to <TO>` - Receiver address
- `--amount <AMOUNT>` - Amount to transfer - `--pretty` - Pretty print amount
(default: false) - `--in-base-unit` - Show amount in base units

fee is-vip-account

Check if an account is a VIP.

```
token-cli fee --name <NAME> is-vip-account --account <ACCOUNT>
```

Options: - `--account <ACCOUNT>` - Account address

fee get-min-fee

Get minimum fee.

```
token-cli fee --name <NAME> get-min-fee [--pretty] [--in-base-unit]
```

Options: - `--pretty` - Pretty print amount (default: false) - `--in-base-unit`
- Show amount in base units

fee get-max-fee

Get maximum fee.

```
token-cli fee --name <NAME> get-max-fee [--pretty] [--in-base-unit]
```

Options: - `--pretty` - Pretty print amount (default: false) - `--in-base-unit`
- Show amount in base units

fee get-current-total-fee-in-pool

Get current total fee in the pool.

```
token-cli fee --name <NAME> get-current-total-fee-in-pool [--pretty] [--in-base-unit]
```

Options: - `--pretty` - Pretty print amount (default: false) - `--in-base-unit`
- Show amount in base units

fee get-total-collected-fee

Get total collected fee so far.

```
token-cli fee --name <NAME> get-total-collected-fee [--pretty] [--in-base-unit]
```

Options: - `--pretty` - Pretty print amount (default: false) - `--in-base-unit`
- Show amount in base units

fee set-fee-settings

Set token fee settings (FEE_ADMIN_ROLE only).

```
token-cli fee --name <NAME> set-fee-settings --enabled <ENABLED> --vip-rate <VIP_RATE> --de
```

Required Options: - `--enabled <ENABLED>` - Enable or disable fee feature
- `--vip-rate <VIP_RATE>` - VIP fee rate (in basis points) - `--default-rate <DEFAULT_RATE>` - Default fee rate (in basis points) - `--min-fee <MIN_FEE>` - Minimum fee (default: 0) - `--max-fee <MAX_FEE>` - Maximum fee (default: 0 for unlimited)

Options: - `--in-base-unit` - Treat min and max fee as base units (true) or full tokens (false, default - more user-friendly)

Task Options: - `--description <DESCRIPTION>` - Description for the set fee settings task - `--min-approvals <COUNT>` - Minimum number of approvals required (default: 1) - `--auto-execute` - Automatically execute after reaching minimum approvals - `--expiry <SECONDS>` - Expiry time in seconds (default: 604800) - `--collect-delay <SECONDS>` - Delay before task collection starts (default: 86400)

fee enable

Enable token fee feature (FEE_ADMIN_ROLE only).

```
token-cli fee --name <NAME> enable [OPTIONS]
```

Task Options: - --description <DESCRIPTION> - Description for the enable fee task - --min-approvals <COUNT> - Minimum number of approvals required (default: 1) - --auto-execute - Automatically execute after reaching minimum approvals - --expiry <SECONDS> - Expiry time in seconds (default: 604800) - --collect-delay <SECONDS> - Delay before task collection starts (default: 86400)

fee disable

Disable token fee feature (FEE_ADMIN_ROLE only).

```
token-cli fee --name <NAME> disable [OPTIONS]
```

Task Options: - --description <DESCRIPTION> - Description for the disable fee task - --min-approvals <COUNT> - Minimum number of approvals required (default: 1) - --auto-execute - Automatically execute after reaching minimum approvals - --expiry <SECONDS> - Expiry time in seconds (default: 604800) - --collect-delay <SECONDS> - Delay before task collection starts (default: 86400)

fee set-account-rate

Set token fee rate for a specific account (FEE_ADMIN_ROLE only).

```
token-cli fee --name <NAME> set-account-rate --account <ACCOUNT> --rate <RATE> [OPTIONS]
```

Required Options: - --account <ACCOUNT> - Account address - --rate <RATE> - Fee rate for the account (in basis points)

Task Options: - --description <DESCRIPTION> - Description for the set account rate task - --min-approvals <COUNT> - Minimum number of approvals required (default: 1) - --auto-execute - Automatically execute after reaching minimum approvals - --expiry <SECONDS> - Expiry time in seconds (default: 604800) - --collect-delay <SECONDS> - Delay before task collection starts (default: 86400)

fee set-vip-account

Set/unset VIP account (FEE_ADMIN_ROLE only).

```
token-cli fee --name <NAME> set-vip-account --account <ACCOUNT> --is-vip <IS_VIP> [OPTIONS]
```

Required Options: - --account <ACCOUNT> - Account address - --is-vip <IS_VIP> - Is VIP or not

Task Options: - `--description <DESCRIPTION>` - Description for the set VIP account task - `--min-approvals <COUNT>` - Minimum number of approvals required (default: 1) - `--auto-execute` - Automatically execute after reaching minimum approvals - `--expiry <SECONDS>` - Expiry time in seconds (default: 604800) - `--collect-delay <SECONDS>` - Delay before task collection starts (default: 86400)

fee withdraw

Withdraw fee from the pool to an authorized fee recipient.

```
token-cli fee --name <NAME> withdraw --to <TO> --amount <AMOUNT> [--in-base-unit]
```

Options: - `--to <TO>` - Recipient address - `--amount <AMOUNT>` - Amount to withdraw - `--in-base-unit` - Treat amount as base units (true) or full tokens (false, default - more user-friendly)

Permission Commands

Commands for managing token permission.

permission settings

Get current status of permission control.

```
token-cli permission settings --name <NAME>
```

Options: - `--name <NAME>` - Token name

permission get-default-permission

Get the default permission applied to all non-private-permission tokens.

```
token-cli permission get-default-permission
```

permission get-token-permission

Get token's effective permission (local or global according to token settings).

```
token-cli permission get-token-permission --name <NAME>
```

Options: - `--name <NAME>` - Token name

permission is-account-locked

Check if an account is locked by a permission object.

```
token-cli permission is-account-locked [--name <NAME>] [--permission <PERMISSION>] --account
```

Options: - `--name <NAME>` - Token name (used to find the effective permission contract) - `--permission <PERMISSION>` - Permission contract address (if not provided, use token's effective permission) - `--account <ACCOUNT>` - Account address - `--force` - Force checking even if token permission is disabled

Note: If both `--name` and `--permission` are provided, `--name` takes precedence. If neither is provided, uses the default permission contract.

permission get-mask

Get permission mask of an account.

```
token-cli permission get-mask [--name <NAME>] [--permission <PERMISSION>] --account <ACCOUNT>
```

Options: - `--name <NAME>` - Token name (used to find the effective permission contract) - `--permission <PERMISSION>` - Permission contract address - `--account <ACCOUNT>` - Account address - `--force` - Force get mask even if token permission is disabled

permission get-kyc-level

Get KYC level of an account.

```
token-cli permission get-kyc-level [--name <NAME>] [--permission <PERMISSION>] --account <ACCOUNT>
```

Options: - `--name <NAME>` - Token name (used to find the effective permission contract) - `--permission <PERMISSION>` - Permission contract address - `--account <ACCOUNT>` - Account address - `--force` - Force get KYC level even if token permission is disabled

permission can-transfer

Check if a sender can transfer an amount of tokens to a receiver.

```
token-cli permission can-transfer --name <NAME> --from <FROM> --to <TO> --amount <AMOUNT>
```

Options: - `--name <NAME>` - Token name - `--from <FROM>` - Sender address - `--to <TO>` - Recipient address - `--amount <AMOUNT>` - Amount to transfer

permission create-permission

Create a permission contract, returns the address of new permission contract.

```
token-cli permission create-permission
```

permission set-permission

Set token's private permission (UPGRADER_ROLE only).

```
token-cli permission set-permission --name <NAME> --permission <PERMISSION> [OPTIONS]
```

Required Options: - `--name <NAME>` - Token name - `--permission <PERMISSION>` - Permission contract address (set to empty string to use default permission)

Task Options: - `--description <DESCRIPTION>` - Description for the set permission task - `--min-approvals <COUNT>` - Minimum number of approvals required (default: 1) - `--auto-execute` - Automatically execute after reaching minimum approvals - `--expiry <SECONDS>` - Expiry time in seconds (default: 604800) - `--collect-delay <SECONDS>` - Delay before task collection starts (default: 86400)

permission enable-permission

Enable permission control (COMPLIANCE_ADMIN_ROLE only).

```
token-cli permission enable-permission --name <NAME> [OPTIONS]
```

Options: - `--name <NAME>` - Token name

Task Options: - `--description <DESCRIPTION>` - Description for the enable permission task - `--min-approvals <COUNT>` - Minimum number of approvals required (default: 1) - `--auto-execute` - Automatically execute after reaching minimum approvals - `--expiry <SECONDS>` - Expiry time in seconds (default: 604800) - `--collect-delay <SECONDS>` - Delay before task collection starts (default: 86400)

permission disable-permission

Disable permission control (COMPLIANCE_ADMIN_ROLE only).

```
token-cli permission disable-permission --name <NAME> [OPTIONS]
```

Options: - `--name <NAME>` - Token name

Task Options: - `--description <DESCRIPTION>` - Description for the disable permission task - `--min-approvals <COUNT>` - Minimum number of approvals required (default: 1) - `--auto-execute` - Automatically execute after reaching minimum approvals - `--expiry <SECONDS>` - Expiry time in seconds (default: 604800) - `--collect-delay <SECONDS>` - Delay before task collection starts (default: 86400)

permission set-mode

Set permission work mode (COMPLIANCE_ADMIN_ROLE only).

```
token-cli permission set-mode --name <NAME> --with-mask <WITH_MASK> --with-kyc <WITH_KYC> [
```

Required Options: - `--name <NAME>` - Token name - `--with-mask <WITH_MASK>` - Enable or disable mask control - `--with-kyc <WITH_KYC>` - Enable or disable KYC control

Task Options: - `--description <DESCRIPTION>` - Description for the set mode task - `--min-approvals <COUNT>` - Minimum number of approvals required (default: 1) - `--auto-execute` - Automatically execute after reaching minimum approvals - `--expiry <SECONDS>` - Expiry time in seconds (default: 604800) - `--collect-delay <SECONDS>` - Delay before task collection starts (default: 86400)

permission lock-account

Lock an account (COMPLIANCE_ADMIN_ROLE only).

```
token-cli permission lock-account --name <NAME> --account <ACCOUNT> [OPTIONS]
```

Required Options: - `--name <NAME>` - Token name - `--account <ACCOUNT>` - Address of account to lock

Task Options: - `--description <DESCRIPTION>` - Description for the lock account task - `--min-approvals <COUNT>` - Minimum number of approvals required (default: 1) - `--auto-execute` - Automatically execute after reaching minimum approvals - `--expiry <SECONDS>` - Expiry time in seconds (default: 604800) - `--collect-delay <SECONDS>` - Delay before task collection starts (default: 86400)

permission unlock-account

Unlock an account (COMPLIANCE_ADMIN_ROLE only).

```
token-cli permission unlock-account --name <NAME> --account <ACCOUNT> [OPTIONS]
```

Required Options: - `--name <NAME>` - Token name - `--account <ACCOUNT>` - Address of account to unlock

Task Options: - `--description <DESCRIPTION>` - Description for the unlock account task - `--min-approvals <COUNT>` - Minimum number of approvals required (default: 1) - `--auto-execute` - Automatically execute after reaching minimum approvals - `--expiry <SECONDS>` - Expiry time in seconds (default: 604800) - `--collect-delay <SECONDS>` - Delay before task collection starts (default: 86400)

permission set-mask

Change the permission mask of an account (COMPLIANCE_ADMIN_ROLE only).

```
token-cli permission set-mask --name <NAME> --account <ACCOUNT> --can-send <CAN_SEND> --can-
```

Required Options: - `--name <NAME>` - Token name - `--account <ACCOUNT>` - Account address - `--can-send <CAN_SEND>` - Can send tokens - `--can-receive <CAN_RECEIVE>` - Can receive tokens

Task Options: - `--description <DESCRIPTION>` - Description for the set mask task - `--min-approvals <COUNT>` - Minimum number of approvals required (default: 1) - `--auto-execute` - Automatically execute after reaching minimum approvals - `--expiry <SECONDS>` - Expiry time in seconds (default: 604800) - `--collect-delay <SECONDS>` - Delay before task collection starts (default: 86400)

permission set-kyc-level

Set KYC level of an account (COMPLIANCE_ADMIN_ROLE only).

```
token-cli permission set-kyc-level --name <NAME> --account <ACCOUNT> --level <LEVEL> [OPTION]
```

Required Options: - `--name <NAME>` - Token name - `--account <ACCOUNT>` - Account address - `--level <LEVEL>` - New KYC level (enum value)

Task Options: - `--description <DESCRIPTION>` - Description for the set KYC level task - `--min-approvals <COUNT>` - Minimum number of approvals required (default: 1) - `--auto-execute` - Automatically execute after reaching minimum approvals - `--expiry <SECONDS>` - Expiry time in seconds (default: 604800) - `--collect-delay <SECONDS>` - Delay before task collection starts (default: 86400)

Role Commands

Commands for managing roles. `--name` option defaults to empty string for global role management.

role is-root

Check if an account is root.

```
token-cli role [--name <NAME>] is-root <ACCOUNT>
```

Arguments: - `ACCOUNT` - Account address to check

Options: - `--name <NAME>` - Token name (empty string for global role management, default: empty)

role get-root

Get the root account.

```
token-cli role [--name <NAME>] get-root
```

Options: - `--name <NAME>` - Token name (empty string for global role management, default: empty)

role is-role

Check if an account has a specific role.

```
token-cli role [--name <NAME>] is-role --account <ACCOUNT> --role <ROLE>
```

Options: - --name <NAME> - Token name (empty string for global role management, default: empty) - --account <ACCOUNT> - Account address to check - --role <ROLE> - Role identifier (enum value)

role get-role-members

Get all members of a role.

```
token-cli role [--name <NAME>] get-role-members --role <ROLE>
```

Options: - --name <NAME> - Token name (empty string for global role management, default: empty) - --role <ROLE> - Role identifier (enum value)

role change-root

Change root account (ROOT only).

```
token-cli role [--name <NAME>] change-root --account <ACCOUNT> [OPTIONS]
```

Required Options: - --account <ACCOUNT> - New root account address

Options: - --name <NAME> - Token name (empty string for global role management, default: empty)

Task Options: - --description <DESCRIPTION> - Description for the change root task - --auto-execute - Automatically execute after reaching minimum approvals - --expiry <SECONDS> - Expiry time in seconds (default: 604800, 0 for no expiry) - --collect-delay <SECONDS> - Delay before task collection starts (default: 86400)

role set-role

Set a role for an account (ROOT only).

```
token-cli role [--name <NAME>] set-role --account <ACCOUNT> --role <ROLE> [OPTIONS]
```

Required Options: - --account <ACCOUNT> - Account address - --role <ROLE> - Role identifier (enum value)

Options: - --name <NAME> - Token name (empty string for global role management, default: empty)

Task Options: - --description <DESCRIPTION> - Description for the set role task - --auto-execute - Automatically execute after reaching minimum approvals - --expiry <SECONDS> - Expiry time in seconds (default: 604800) - --collect-delay <SECONDS> - Delay before task collection starts (default: 86400)

role unset-role

Unset a role for an account (ROOT only).

```
token-cli role [--name <NAME>] unset-role --account <ACCOUNT> --role <ROLE> [OPTIONS]
```

Required Options: - --account <ACCOUNT> - Account address - --role <ROLE> - Role identifier (enum value)

Options: - --name <NAME> - Token name (empty string for global role management, default: empty)

Task Options: - --description <DESCRIPTION> - Description for the unset role task - --auto-execute - Automatically execute after reaching minimum approvals - --expiry <SECONDS> - Expiry time in seconds (default: 604800) - --collect-delay <SECONDS> - Delay before task collection starts (default: 86400)

Common Task Options

Many commands that create tasks support the following common options:

- --min-approvals <COUNT> - Minimum number of approvals required (default: 1)
- --auto-execute - Automatically execute after task reaching minimum approvals
- --expiry <SECONDS> - Expiry time in seconds (default: 604800 = 7 days, 0 for no expiry)
- --collect-delay <SECONDS> - Delay in seconds before task collection starts (default: 86400 = 1 day, 0 for no collection, 1 for immediate collection)
- --description <DESCRIPTION> - Description for the task

Role Types

Available role types (as CLI argument values): - ROOT - Root account with highest privileges - DEFAULT_ADMIN_ROLE - Default admin role - UPGRADER_ROLE - Upgrader role for contract upgrades - MINT_ADMIN_ROLE - Mint admin role for minting tokens - BURN_ADMIN_ROLE - Burn admin role for burning tokens - COMPLIANCE_ADMIN_ROLE - Compliance admin role for KYC and permissions - FEE_ADMIN_ROLE - Fee admin role for managing fees - FEE_RECIPIENT_ROLE - Fee recipient role for receiving fees - PAUSER_ROLE - Pauser role for freezing/unfreezing tokens

KYC Levels

Available KYC level types: - KYC0 - No KYC verification - KYC1 - Basic KYC - KYC2 - Level 2 KYC - KYC3 - Level 3 KYC - KYC4 - Full KYC verification

Notes

1. **Task Lifecycle:** Many administrative operations create tasks that require approval before execution. Tasks go through states: Pending → Ready → Executed (or Expired/Aborted).
2. **Expiry Time:** Tasks have an expiry time (default 7 days). If not approved within this time, they expire and cannot be executed.
3. **Collection Delay:** After a task reaches a terminal state, it remains in the system for a grace period (default 1 day) before being collected (removed).
4. **Auto Execute:** When enabled, tasks will automatically execute once they reach the minimum approval threshold.
5. **Remote vs Local Operations:** Utility commands support both remote (API-based) and local (client-side) execution. Local operations may produce different results due to implementation differences.
6. **Private Keys:** Always handle private keys securely. The CLI supports both direct private key input and encrypted keystore wallets.