

GenLayer CLI Reference

Each command includes syntax, usage information, and examples to help you effectively use the CLI for interacting with the GenLayer environment.

Command line syntax

General syntax for using the GenLayer CLI:

```
genlayer command [command options] [arguments...]
```

Version Compatibility

The GenLayer CLI always uses the latest compatible version of the environment, ensuring that you benefit from the most recent features, bug fixes, and optimizations without requiring manual updates. If a specific version is needed, you can specify it using the `--localnet-version` option when initializing the environment.

```
genlayer init --localnet-version v0.10.2
```

Commands and usage

Initialize

Prepares and verifies your environment to run the GenLayer Studio.

```
USAGE:  
  genlayer init [options]
```

```
OPTIONS:
```

```
--numValidators <numValidators>      Number of validators (default: "5")
--headless                             Headless mode (default: false)
--reset-db                             Reset Database (default: false)
--localnet-version <localnetVersion>  Select a specific localnet version
```

EXAMPLES:

```
genlayer init
genlayer init --numValidators 10 --headless --reset-db --localnet-version v0.10.2
```

Start GenLayer environment

accounts.

USAGE:

```
genlayer up [options]
```

OPTIONS:

```
--reset-validators      Remove all current validators and create new ran
--numValidators <numValidators>  Number of validators (default: "5")
--headless              Headless mode (default: false)
--reset-db              Reset Database (default: false)
```

EXAMPLES:

```
genlayer up
genlayer up --reset-validators --numValidators 8 --headless --reset-db
```

Stop GenLayer environment

Stops all running GenLayer Localnet services.

USAGE:

```
genlayer stop
```

Create a New GenLayer Project

Initialize a new GenLayer project using a local template.

USAGE:

```
genlayer new <projectName> [options]
```

OPTIONS:

```
--path <directory> Specify the directory for the new project (default: ".")  
--overwrite          Overwrite existing directory if it exists (default: false)
```

EXAMPLES:

```
genlayer new myProject  
genlayer new myProject --path ./customDir  
genlayer new myProject --overwrite
```

Manage CLI Configuration

Configure the GenLayer CLI settings.

USAGE:

```
genlayer config <command> [options]
```

COMMANDS:

```
set <key=value> Set a configuration value  
get [key]       Get the current configuration  
reset <key>     Reset a configuration value to its default
```

EXAMPLES:

```
genlayer config get  
genlayer config get defaultOllamaModel  
genlayer config set defaultOllamaModel=deepseek-r1  
genlayer config reset keyPairPath
```

Deploy and Call Intelligent Contracts

Deploy and interact with intelligent contracts.

USAGE:

```
genlayer deploy [options]  
genlayer call <contractAddress> <method> [options]  
genlayer write <contractAddress> <method> [options]  
genlayer schema <contractAddress> [options]
```

OPTIONS (deploy):

```
--contract <contractPath> (Optional) Path to the intelligent contract to deploy  
--args <args...>           Positional arguments for the contract (space-separated)  
--rpc <rpcUrl>             RPC URL for the network
```

```

OPTIONS (call):
  --args <args...>      Positional arguments for the method (space-separated,
  --rpc <rpcUrl>        RPC URL for the network

OPTIONS (write):
  --args <args...>      Positional arguments for the method (space-separated,
  --rpc <rpcUrl>        RPC URL for the network

OPTIONS (schema):
  --rpc <rpcUrl>        RPC URL for the network

EXAMPLES:
  genlayer deploy
  genlayer deploy --contract ./my_contract.gpy
  genlayer call 0x123456789abcdef greet --args "Hello World!"
  genlayer write 0x123456789abcdef setValue --args 42 "New Value"
  genlayer schema 0x123456789abcdef

```

Deploy Behavior

- If `--contract` is specified, the command will **deploy the given contract**.
- If `--contract` is omitted, the CLI will **search for scripts inside the `deploy` folder**, sort them, and execute them sequentially.

Call vs Write

- **call**: Read-only operations that don't modify contract state.
- **write**: Operations that modify contract state and require transaction confirmation.

Schema Command

- Retrieve the contract schema showing all available methods and their parameters.

Transaction Management

Manage and query transaction receipts and appeals.

```

USAGE:
  genlayer receipt <txId> [options]
  genlayer appeal <txId> [options]

```

```

OPTIONS (receipt):
  --status <status>      Transaction status to wait for (default: "FINALIZED")
                          Valid values: UNINITIALIZED, PENDING, PROPOSING, COMMITTI
                          REVEALING, ACCEPTED, UNDETERMINED, FINALIZED, CANCELED,
                          APPEAL_REVEALING, APPEAL_COMMITTING, READY_TO_FINALIZE,
                          VALIDATORS_TIMEOUT, LEADER_TIMEOUT
  --retries <retries>    Maximum number of retry attempts (default: 100)
  --interval <interval>  Interval between retries in milliseconds (default: 5000)
  --rpc <rpcUrl>         RPC URL for the network

OPTIONS (appeal):
  --rpc <rpcUrl>         RPC URL for the network

EXAMPLES:
  genlayer receipt 0x1234567890abcdef
  genlayer receipt 0x1234567890abcdef --status ACCEPTED
  genlayer receipt 0x1234567890abcdef --status FINALIZED --retries 50 --interval 30
  genlayer receipt 0x1234567890abcdef --rpc http://localhost:8545

  genlayer appeal 0x1234567890abcdef
  genlayer appeal 0x1234567890abcdef --rpc http://localhost:8545

```

Network Management

Configure and view network settings.

```

USAGE:
  genlayer network set [network]
  genlayer network info
  genlayer network list

COMMANDS:
  set [network]  Set the network (interactive menu if network not specified)
  info          Show current network configuration and contract addresses
  list         List all available networks

EXAMPLES:
  genlayer network set           # Interactive selection menu
  genlayer network set localnet  # Set network to localnet
  genlayer network set testnet-asimov # Set network to testnet-asimov
  genlayer network info         # Show current network details
  genlayer network list        # List available networks

```

Account Management

Create and manage accounts with encrypted keystores.

USAGE:

```
genlayer account create [options]
genlayer account show [options]
genlayer account list
genlayer account use <name>
genlayer account remove <name>
genlayer account lock [options]
genlayer account unlock [options]
genlayer account export [options]
genlayer account import [options]
genlayer account send [options]
```

COMMANDS:

create	Create a new account with encrypted keystore
show	Show account details and balance
list	List all accounts
use	Set an account as active
remove	Remove an account
lock	Lock an account (remove cached private key)
unlock	Unlock an account (cache private key in OS keychain)
export	Export account to standard web3 keystore format
import	Import account from keystore file or private key
send	Send GEN tokens to another address

OPTIONS (create):

--name <name>	Account name (default: "default")
--overwrite	Overwrite existing account

OPTIONS (show):

--account <name>	Account to show (default: active account)
--rpc <rpcUrl>	RPC URL for balance lookup

OPTIONS (lock/unlock):

--account <name>	Account to lock/unlock
------------------	------------------------

OPTIONS (export):

--account <name>	Account to export
--output <path>	Output file path
--password <password>	Password for exported keystore

OPTIONS (import):

--name <name>	Name for imported account
--keystore <path>	Path to keystore file
--private-key <key>	Private key to import (alternative to keystore)

OPTIONS (send):

```
--to <address>      Recipient address
--amount <amount>   Amount to send (e.g., "1.5" or "1.5gen")
--account <name>    Account to send from
```

EXAMPLES:

```
genlayer account create --name owner
genlayer account show
genlayer account show --account operator
genlayer account list
genlayer account use owner
genlayer account unlock --account owner
genlayer account export --account operator --output ./operator-keystore.json
genlayer account import --name backup --keystore ./backup.json
genlayer account send --to 0x123... --amount 100gen
```

Staking Operations

Staking operations for validators and delegators on testnet.

USAGE:

```
genlayer staking wizard [options]
genlayer staking validator-join [options]
genlayer staking validator-deposit [options]
genlayer staking validator-exit [options]
genlayer staking validator-claim [options]
genlayer staking validator-prime [options]
genlayer staking validator-info [options]
genlayer staking set-operator [options]
genlayer staking set-identity [options]
genlayer staking delegator-join [options]
genlayer staking delegator-exit [options]
genlayer staking delegator-claim [options]
genlayer staking delegation-info [options]
genlayer staking epoch-info [options]
genlayer staking active-validators [options]
genlayer staking quarantined-validators [options]
genlayer staking banned-validators [options]
```

COMMANDS:

```
wizard          Interactive wizard to become a validator
validator-join  Join as a validator by staking tokens
validator-deposit  Make an additional deposit as a validator
validator-exit  Exit as a validator by withdrawing shares
validator-claim  Claim validator withdrawals after unbonding period
validator-prime  Prime a validator to prepare stake for next epoch
validator-info  Get information about a validator
set-operator    Change the operator address for a validator
set-identity    Set validator identity metadata (moniker, website, socials)
```

```
delegator-join      Join as a delegator by staking with a validator
delegator-exit     Exit as a delegator by withdrawing shares
delegator-claim    Claim delegator withdrawals after unbonding period
delegation-info    Get delegation info for a delegator with a validator
epoch-info         Get current epoch and staking parameters
active-validators  List all active validators
quarantined-validators List all quarantined validators
banned-validators  List all banned validators
```

OPTIONS (wizard):

```
--account <name>      Account to use (skip selection)
--network <network>   Network to use (skip selection)
--skip-identity       Skip identity setup step
```

OPTIONS (validator-join):

```
--amount <amount>     Amount to stake (e.g., "42000gen")
--operator <address>  Operator address (defaults to signer)
--account <name>     Account to use
```

OPTIONS (validator-deposit):

```
--validator <address> Validator wallet address (required)
--amount <amount>     Amount to deposit
```

OPTIONS (validator-exit):

```
--validator <address> Validator wallet address (required)
--shares <shares>     Number of shares to withdraw
```

OPTIONS (validator-claim):

```
--validator <address> Validator wallet address (required)
```

OPTIONS (validator-prime):

```
--validator <address> Validator address to prime
```

OPTIONS (set-operator):

```
--validator <address> Validator wallet address
--operator <address> New operator address
```

OPTIONS (set-identity):

```
--validator <address> Validator wallet address
--moniker <name>     Validator display name (required)
--website <url>     Website URL
--description <text> Description
--twitter <handle>  Twitter handle
--github <handle>   GitHub handle
```

OPTIONS (delegator-join):

```
--validator <address> Validator to delegate to
--amount <amount>    Amount to stake
```

OPTIONS (delegator-exit):

```
--validator <address> Validator to exit from
--shares <shares>    Number of shares to withdraw
```

```

OPTIONS (delegation-info):
  --validator <address>      Validator address
  --delegator <address>     Delegator address (defaults to signer)

COMMON OPTIONS:
  --network <network>       Network to use (localnet, testnet-asimov)
  --rpc <rpcUrl>           RPC URL for the network
  --staking-address <addr>  Override staking contract address

EXAMPLES:
# Become a validator using the interactive wizard
genlayer staking wizard

# Join as validator manually
genlayer staking validator-join --amount 42000gen --operator 0x123...

# Check validator status
genlayer staking validator-info

# Set validator identity
genlayer staking set-identity --validator 0x123... --moniker "My Validator" --web

# Delegate to a validator
genlayer staking delegator-join --validator 0x123... --amount 1000gen

# Check your delegation info
genlayer staking delegation-info --validator 0x123...

# Check epoch info
genlayer staking epoch-info

# List active validators
genlayer staking active-validators

```

Account Management

View and manage your account.

```

USAGE:
  genlayer account          Show account info (address, balance, network,
  genlayer account create [options]  Create a new account
  genlayer account import [options]  Import an account from a private key
  genlayer account send <to> <amount>  Send GEN to an address
  genlayer account unlock          Unlock account (cache key in OS keychain)
  genlayer account lock           Lock account (remove key from OS keychain)

```

```
OPTIONS (create):
  --output <path>    Path to save the keystore (default: "./keypair.json")
  --overwrite        Overwrite existing file (default: false)

OPTIONS (import):
  --private-key <key> Private key to import (will prompt if not provided)
  --output <path>    Path to save the keystore (default: "./keypair.json")
  --overwrite        Overwrite existing file (default: false)

EXAMPLES:
  genlayer account
  genlayer account create
  genlayer account create --output ./my_key.json --overwrite
  genlayer account import
  genlayer account import --private-key 0x1234...abcd
  genlayer account import --private-key 0x1234...abcd --output ./imported.json
  genlayer account send 0x123...abc 10gen
  genlayer account send 0x123...abc 0.5gen
  genlayer account unlock
  genlayer account lock
```

Update Resources

Manage and update models or configurations.

USAGE:

```
genlayer update ollama [options]
```

OPTIONS:

```
--model [model-name] Specify the model to update or pull
--remove              Remove the specified model instead of updating
```

EXAMPLES:

```
genlayer update ollama
genlayer update ollama --model deepseek-r1
genlayer update ollama --model deepseek-r1 --remove
```

Localnet Validator Management

Manage localnet validator operations.

USAGE:

```
genlayer localnet validators <command> [options]
```

COMMANDS:

```
get [--address <validatorAddress>] Retrieve details of a specific validator o
delete [--address <validatorAddress>] Delete a specific validator or all validat
count Count all validators
update <validatorAddress> [options] Update a validator details
create-random [options] Create random validators
create [options] Create a new validator
```

OPTIONS (update):

```
--stake <stake> New stake for the validator
--provider <provider> New provider for the validator
--model <model> New model for the validator
--config <config> New JSON config for the validator
```

OPTIONS (create-random):

```
--count <count> Number of validators to create (default: "
--providers <providers...> Space-separated list of provider names (e.
--models <models...> Space-separated list of model names (e.g.,
```

OPTIONS (create):

```
--stake <stake> Stake amount for the validator (default: "
--config <config> Optional JSON configuration for the valida
--provider <provider> Specify the provider for the validator
--model <model> Specify the model for the validator
```

EXAMPLES:

```
genlayer localnet validators get
genlayer localnet validators get --address 0x123456789abcdef

genlayer localnet validators count
genlayer localnet validators delete --address 0x123456789abcdef
genlayer localnet validators update 0x123456789abcdef --stake 100 --provider open

genlayer localnet validators create
genlayer localnet validators create --stake 50 --provider openai --model gpt-4
genlayer localnet validators create-random --count 3 --providers openai --models
```

Manage staking for validators and delegators on testnet-asimov.

USAGE:

```
genlayer staking <command> [options]
```

VALIDATOR COMMANDS:

validator-join	Join as a validator by staking tokens
validator-deposit	Make an additional deposit as a validator
validator-exit	Exit as a validator by withdrawing shares
validator-claim	Claim validator withdrawals after unbonding period
validator-prime	Prime a validator to prepare for next epoch
validator-info	Get information about a validator
set-operator	Change the operator address for a validator
set-identity	Set validator identity metadata

DELEGATOR COMMANDS:

delegator-join	Join as a delegator by staking with a validator
delegator-exit	Exit as a delegator by withdrawing shares
delegator-claim	Claim delegator withdrawals after unbonding period
stake-info	Get stake info for a delegator with a validator

INFO COMMANDS:

epoch-info	Get current epoch and staking parameters
active-validators	List all active validators
quarantined-validators	List all quarantined validators
banned-validators	List all banned validators

OPTIONS (validator-join):

--amount <amount>	Stake amount (e.g., "42000gen")
--operator <address>	Optional operator address for the validator
--staking-address <addr>	Override staking contract address

OPTIONS (delegator-join):

--validator <address>	Validator address to delegate to (required)
--amount <amount>	Amount to stake (e.g., "100gen")
--staking-address <addr>	Override staking contract address

OPTIONS (validator-exit/validator-claim):

--validator <address>	Validator wallet address (required)
--shares <shares>	Number of shares to withdraw (exit only)

OPTIONS (delegator-exit/delegator-claim):

--validator <address>	Validator address (required)
--shares <shares>	Number of shares to withdraw (exit only)
--staking-address <addr>	Override staking contract address

OPTIONS (set-identity):

--validator <address>	Validator address (required)
--moniker <name>	Display name (required)
--website <url>	Website URL
--description <text>	Description
--twitter <handle>	Twitter handle
--telegram <handle>	Telegram handle
--github <handle>	GitHub handle
--email <email>	Contact email
--logo-uri <uri>	Logo image URL

EXAMPLES:

```
# Join as validator with 42000 GEN stake
genlayer staking validator-join --amount 42000gen

# Join with separate operator address (recommended for security)
genlayer staking validator-join --amount 42000gen --operator 0xOperator...

# Set validator identity
genlayer staking set-identity --validator 0x... --moniker "My Validator" --website

# Add more stake to validator
genlayer staking validator-deposit --validator 0xValidatorWallet... --amount 1000

# Check epoch info (includes inflation data)
genlayer staking epoch-info

# Get validator info
genlayer staking validator-info --validator 0x...

# Delegate to a validator
genlayer staking delegator-join --validator 0x... --amount 100gen

# Check stake info (includes projected rewards)
genlayer staking stake-info --validator 0x...

# Validator exit and claim (requires validator wallet address)
genlayer staking validator-exit --validator 0xValidatorWallet... --shares 100
genlayer staking validator-claim --validator 0xValidatorWallet...

# Delegator exit and claim (7 epoch unbonding period)
genlayer staking delegator-exit --validator 0x... --shares 50
genlayer staking delegator-claim --validator 0x...

# List active validators
genlayer staking active-validators
```

Last updated on December 5, 2025