VotingEscrow

Users create and control locks on theVotingEscrow contract. A user may only have one lock at any given time. Adding to a lock resets the 5-year lock period.

The code samples below demonstrate how to manage locks using ethers.js or foundry. You can also manage locks using a block explorer likeoklink.com/canto .

Creating a Lock

To create a lock, call thecreateLock(uint256 _value) payable method. The call value and_value parameter must match.

ethers.js

```

Copy amount = ethers.utils.parseEther("100") // 100 CANTO await VotingEscrow.createLock(amount, { value: amount })

```

foundry

```

Copy cast send --ledger 0x... "createLock(uint256)" 100 --value 100ether

```

Adding to a Lock

To add to your existing lock, call theincreaseAmount(uint256 _value) payable method. The call value and_value parameter must match.

ethers.js

```

Copy amount = ethers.utils.parseEther("100") // 100 CANTO await VotingEscrow.increaseAmount(amount, { value: amount })

```

foundry

```

Copy cast send --ledger 0x... "increaseAmount(uint256)" 100 --value 100ether

```

Reading Voting Power

To read voting power, call thebalanceOf(address _owner) view.

ethers.js

```

Copy const votingPower = await VotingEscrow.balanceOf("0x...")

```

foundry

```

Copy cast call 0x... "balanceOf(address)" 0x...

```

Withdrawing

To withdraw a completed lock, call thewithdraw() method.

ethers.js

```

Copy await VotingEscrow.withdraw()

```

foundry

```

Copy cast send --ledger 0x... "withdaw()"

```

Previous Liquidity Coordinator Next GaugeController Last updated1 month ago On this page *Creating a Lock * Reading Voting Power * Withdrawing