

Instructions

1. Download Lotus Binary

1. Open the [Lotus v1.31.0 Release](#) page in your web browser.
2. Scroll down to the [Assets](#) section and download the appropriate Lotus binary for your operating system.

2. Extract the Archive

1. Once the download is complete, extract the content of the archive to a desired location.

3. Navigate to the Lotus Directory

1. Open a terminal or command prompt.
2. Navigate to the folder where the extracted Lotus binary is located.

4. Start the Lotus Daemon

Run the following command in the terminal:

```
env FULLNODE_API_INFO=wss://wss.node.glif.io/apigw/lotus ./lotus daemon --lite
```

5. Modify the Configuration File

1. Open the `~/.lotus/config.toml` file.
2. In the `[Wallet]` section, remove the `#` symbol before `EnableLedger` and set the value to `true`:

```
[Wallet]
# type: string
# env var: LOTUS_WALLET_REMOTEBACKEND
# RemoteBackend = ""
# type: bool
# env var: LOTUS_WALLET_ENABLELEDGER
EnableLedger = true

# type: bool
# env var: LOTUS_WALLET_DISABLELOCAL
# DisableLocal = false
```

6. Restart the Lotus Daemon

1. Stop the Lotus daemon by pressing **Ctrl+C** in the terminal.
2. Run the Lotus daemon again.

7. Open a New Terminal Window

1. Navigate to the folder where the extracted Lotus binary is located.
2. Keep the first terminal running the Lotus daemon, and use the second terminal to enter commands.

8. Prepare Your Ledger Device

1. Unlock your Ledger device.
2. Open the Filecoin app on the Ledger and ensure it remains connected to your computer.

9. Create a Ledger-Backed Wallet

Use the following command to create a new Ledger-backed wallet (**secp256k1-ledger**):

```
./lotus wallet new secp256k1-ledger
```

On your Ledger device, confirm the creation of the wallet.

Keep running the command until you see the ledger wallet addresses that you use to sign the multisig transaction.

Copy the response from the command as it will be needed later to create the Lotus multisig propose.

10. Prepare calldata

1. Visit abi.hashex.org.
2. Manually enter your parameters as required:
 - For Function, select your function and set **decreaseAllowance** as a function name.

- [illegible]

3. Copy the Encoded data.

Function

 \wedge

decreaseAllowance

 \wedge

×

 \wedge

×

[illegible]

1. Visit cbor.me.
2. On the left side, enter the copied encoded data:

h'encoded data'

3. Select the checkbox plain text, then click Convert to bytes.
4. Copy the converted data from the right side.

[illegible]

12. Create Lotus multisig propose

Run the following command to create a multisig proposal:

lotus msig propose --from=proposerAddress walletAddress destinationAddress value

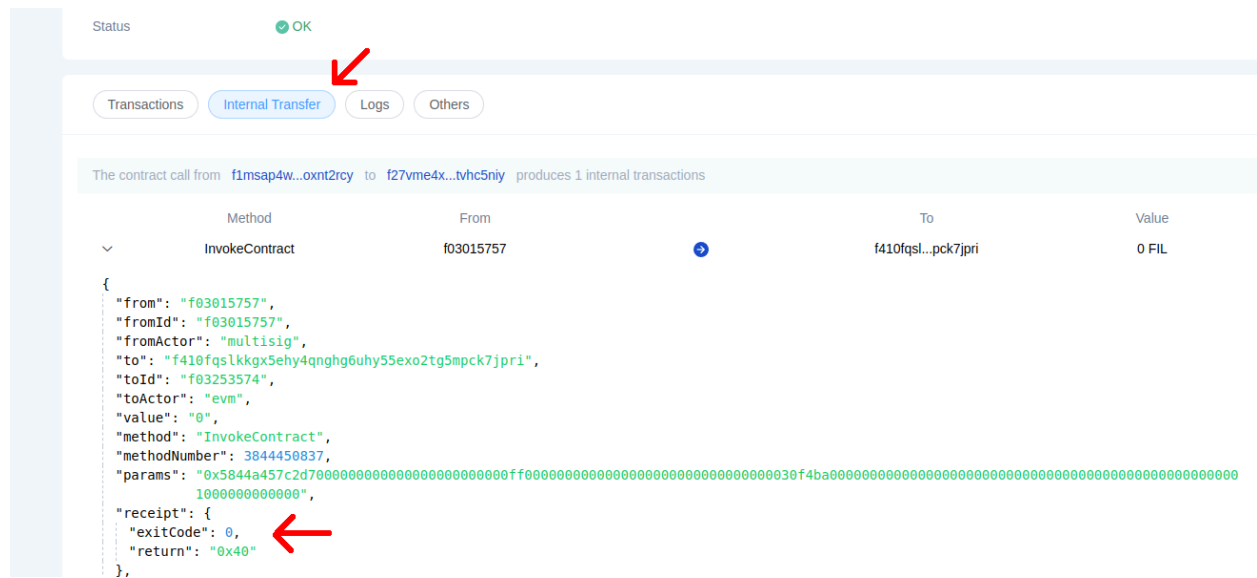
Where:

- `proposerAddress` is the address of your ledger wallet.
- `walletAddress` is the address of the allocator multisig.
- `destinationAddress` is the **Client Contract Address** from the client's JSON.
- `value` is the converted data from CBOR.

Executing the command will generate a transaction CID, which can be used to verify the transaction on Filfox. If the multisig requires more than one approval, log in to your multisig account, complete the signing process, and then check the returned CID by multisig on Filfox to confirm the transaction's success.

https://filfox.info/en/message/<returned_cid>

Once the transaction is completed, go to the Internal Transfer section, expand InvokeContract, and check if `exitCode` equals 0. If not, an error has occurred at some stage.



Important Information

Removing DataCap does not directly return it to the allocator but remains on the contract. To utilize this DataCap, you must repeat the steps from `Prepare calldata`, but instead of setting `decreaseAllowance`, set `increaseAllowance`.