

# Exchange integration

This page lists the general steps and workflows you need to follow to offer FIL on an exchange.

## Running a Filecoin node

If you plan to offer FIL on your exchange, you will need to run a Filecoin node. [Lotus](#) is the reference implementation node for the Filecoin network, and as such, is currently the most production-ready implementation available.

## Node setup and installation

Follow the [Lotus installation guide](#) to properly install the Lotus applications and launch node. The basic steps are:

- Prepare your hardware by meeting the minimal requirements.
- Install dependencies.
- Start the Lotus daemon and sync the chain by either:
  - - Syncing from scratch: in your lotus directory run `lotus daemon`
  - - Syncing from a full snapshot.
  - - Syncing from minimal snapshot
- \*
- 

A snapshot only has the state trees from the recent tipset (2000 epochs) onward and nothing before that tipset, which means it does not have all the historical states of the network. In addition, only a full snapshot has full state trees from a certain tipset.

## Check sync status

You can check the sync status by running `lotus sync status`. You are fully synced when the Height difference is 0. Lotus will output any sync error.

...

## Copy lotus sync status

```
worker 28975: Base: [bafy2bzacec2exqtdvxzzvikwg3sg67e57yaocpmrljkfxth3isjmrzud45vsa
bafy2bzacecbbqdvidaf56eeth7oul2raycjpddvgtb23ywoyrqyh7ajulepai
bafy2bzaceajtc3mgie5b72ivuiv3svwnbfq7kl2gnbni5tqc2ja6zi22vkttu
bafy2bzacebf7oavffgiajf5goi6verhepsclhb7pcfwiwkp4dxdqshadw
bafy2bzaceddymcr5onnh63uznctks5hujpgzxcd45f5ef7oiuqocpmqs6rh2
bafy2bzacebuqm43mspki4ekdbu6xj663mpkrcoxdss64t27fo77bg7pzbaia
bafy2bzaceaxurgudmnpjhcoz5sfad7yd63axyhmretqr5jjmvnicyu7betvvi
bafy2bzaceali3oihrslxmnjdu2ysesxradorczjimruxdbkqidtcuag7wlr5a] Target:
[bafy2bzacedmswd73yn5faqzpbu5ofqinmh6fnzxfvazy52jrfthfhod4ssfgw] (1272951) Height diff: 1 Stage: complete
Height: 1272951 Elapsed: 243.770396ms worker 28976: Base:
[bafy2bzacedmswd73yn5faqzpbu5ofqinmh6fnzxfvazy52jrfthfhod4ssfgw] Target:
[bafy2bzacebxf27yxmzksno5ajkvp6shwojrx5mxsrmlchrdchaqfg2wo7o3i] (1272951) Height diff: 0 Stage:
complete Height: 1272951 Elapsed: 1.00020144s worker 28977: Base: [] Target: [] (0) Height diff: 0 Stage: idle
Height: 0
```

...

You can run `lotus sync wait` to wait for the sync to be complete. Lotus will output `Done!` once your node is fully synced.

## Basic network technology info

The Filecoin network uses a [Proof of Storage \(PoRep\)](#) + [Proof of SpaceTime \(PoSt\)](#) consensus algorithm. Time in the Filecoin network is dissected into [epochs](#) set to 30 seconds. A new set of blocks is produced for every epoch for a [tipset](#). The hard finality of the Filecoin network is 900 epochs.

## Accounts and wallets

Filecoin uses an account-based model. There are 4 types of account prefixes:

- f0
- for ID address

- f1
- for Secp256k1 wallets
- f2
- for [actor](#)
- accounts
- f3
- for BLS wallets
- 

f1 ,f2 , and f3 prefixed addresses are called account addresses. An account address is activated when it first receives a transaction. f0 prefixed addresses are mapped to each active account address.

Testnet addresses Within a testnet, the address prefix is t. So ID addresses become t0 , Secp256k1 wallets become t1 , etc.

## Signatures

Filecoin currently uses two types of signatures:

- ECDSA signatures over the Secp256k1 elliptic curve
- BLS signatures over the BLS12-381 group of curves.
- 

Details and reference implementations can be found [in the Filecoin specification](#) .

## Messages

There are two message types:

- [Signed messages](#)
- [Unsigned messages](#)
- .
- 

Messages are fully irreversible at 900 epochs. Waiting 200 epochs for message confirmation is acceptable.

There are multiple gas fees associated with each message. Refer to the [practical guide to gas section of this blog post](#) for details.

An ExitCode of 0 in the message receipt indicates that the message was sent successfully.

## Mempool

When a user sends a transaction to the network, it gets placed into the mempool queue. If a transaction doesn't have enough gas, it stays in the mempool and doesn't go anywhere. To new users, it looks like this transaction is lost forever. However, users can update the transaction with an updated GasLimit , GasFeeCap , and/or GasPremium . As long as you don't change anything else in the transaction (nonce , to , from , value ), then the transaction that is sat in the mempool will get updated with the new gas allowance.

## Expiration

There is no limit for how long a message can spend in the mempool. However, the mempool does get cleaned when there are too many messages in it, starting with the messages with the least gas.

## Automatic gas values

When GasFeeCap , GasPremium and MaxFee are set to 0 , Lotus will do the gas estimation for the message with 25% overestimation for the gas limit based on the current network condition.

Some JavaScript libraries attempt to estimate the gas fees before sending the transaction to the Filecoin network. However, they sometimes underestimate, leading to transactions getting stuck in the mempool. If you are noticing your transactions getting stuck in the mempool after sending them to the network using a JavaScript library, try GasFeeCap , GasPremium , and MaxFee to 0 .

## Integration

You can interact with the network by using Lotus CLI or using the [JSON-RPC APIs](#) . Follow the [API tokens guide](#) to set up API tokens on your node and grant necessary permissions. To find all CLI usage, run `lotus -h` in your lotus folder.

You can find some other API client libraries developed by the Filecoin community [within the API client libraries page](#) .

## API examples

## ChainHead

...

...

...

## ChainGetParentMessages

...

...

WalletNew

...

...

WalletBalance

...

...

## MpoolPushMessage

" " "

" " "

Method ID of 0 with null Params is a balance transfer transaction. When the GasFeeCap, GasPremium and MaxFee is 0, Lotus will do the gas estimation for the message with a 25% overestimation for the gas limit based on the current network condition. You can change this value via the GasLimitOverestimation field.

## FAQ

How do I sign a message?

Use [WalletSign](#) to sign the message and send the signed message using [MpoolPush](#).

You may also use this [Filecoin signing tool library](#).

How do I retrieve the gas fees of a message?

Call [StateReplay](#) and look up the GasCost in the response.

How to get the gas estimation of a message?

You can estimate the gas cost of a message by calling [GasEstimateMessageGas](#). This API estimates the gas limit with a 25% overestimation based on the network condition under the given tipset key. You can change this value via the GasLimitOverestimation field.

How do I ensure that all balances transfer in any messages are captured, including msig transfers?

Call [StateCompute](#) or [StateReplay](#) and go through all the transactions in the execution trace. Whenever the value != 0 && exit code == 0, it indicates a balance transfer has occurred.

How can I check if my transaction is stuck?

The Lotus RPC method to retrieve the list of transactions waiting on the mempool is [Filecoin.MpoolPending](#). The RPC call is:

...

```
Copy { "jsonrpc": "2.0", "method": "Filecoin.MpoolPending", "id": 1, "params": [null] }
```

...

If you are using a JavaScript library, the method you need is [mempoolPending](#).

Join the Filecoin Slack

Join the [Filecoin Slack](#) and post any questions you have in there.

## Useful Links

- [Filecoin website](#)
- [Filecoin Specs](#)
- [Filecoin Networks](#)
- [Lotus](#)
- - the reference implementation for the Filecoin network
- [Block Explorers](#)
- 

[Previous Exchanges](#) [Next Built-in actors](#)

Last updated 5 months ago