Call built-in actors

Filecoin built-in actors can be invoked in a smart contract using either the Protocol API or the Zondax filecoin.solifity library. This page provides instructions on how to use each method.

For conceptual information on built-in actors, including their purposes, how they work and available types, see theconceptual guide Built-in actors can be invoked using the ProtocolJSON-RPC API or the Zondaxfilecoin.sol API.

APIs compared

The ProtocolJSON-RPC API:

- Is maintained by Protocol Labs (PL).
- Uses JSON-RPC, a standardized way to encode remote procedure calls in JSON that can be transported using HTTP or WebSockets.
- Provides a language agnostic interface for Filecoin functionality.
- Allows applications to access Filecoin functionality using HTTP or WebSockets calls to a Filecoin node, like the Lotus daemon.
- Requires authentication for some API calls.
- Serves as the foundation for language-specific libraries (some of which are maintained by organizations other than PL) such asfilecoin.js
- .
-

The Zondaxfilecoin.sol API:

- Supportssome but not all of the built-in actors and their methods
- .
-

Protocol API

Smart contracts can directly access built-in actors and methods using the Protocol API. Links to the reference guides for each of the available actor methods is listed below:

- Account actor
- Datacap
- Miner
- Multisig
- Storage market actor
- Storage power actor
- Verified registry actor
-

Filecoin.sol

Smart contracts can access built-in actor methods with thefilecoin.sol library, a set of Solidity libraries that allow Solidity smart contracts to seamlessly call methods of Filecoin built-in actors. Thefilecoin.sol library supports cross-platform calls to real Filecoin built-in actors. This section contains information on the actors and methods available fromfilecoin.sol , along with installation instructions and working examples of smart contracts that call built-in actor methods.

To invoke built-in actor methods usingfilecoin.sol , follow these steps:

1. Review theavailable actors and methods
2. .
3. Importfilecoin.sol
4. .
5. Call a built-in actor
6. .
7.

Available actors and methods

The majority of the Account, DataCap, Storage Market, Miner, Storage Owner and Verified Registry actor methods are supported and are listed below.Cron, Payment Channel, Reward and System actor methods are currently not supported.

Account

Method Supported? AuthenticateMessage ✔ Constructor ✖ PubkeyAddress ✖ UniversalReceiverHook ✔

DataCap

Method Supported? Allowance ✔ BalanceOf ✔ Burn ✔ BurnFrom ✔ Constructor ✖ DecreaseAllowance ✔ Destroy ✖ IncreaseAllowance ✔ Mint ✖ Name ✔ RevokeAllowance ✔ Symbol ✔ TotalSupply ✔ Transfer ✔ TransferFrom ✔

Miner

Method Supported? ApplyRewards ✖ ChangeBeneficiary ✔ ChangeMultiaddrs ✔ ChangeOwnerAddress ✔ ChangePeerID ✔ ChangeWorkerAddress ✔ CheckSectorProven ✖ CompactPartitions ✖ CompactSectorNumbers ✖ ConfirmSectorProofsValid ✖ ConfirmUpdateWorkerKey ✖ Constructor ✖ ControlAddresses ✖ DeclareFaults ✖ DeclareFaultsRecovered ✖ DisputeWindowedPoSt ✖ ExtendSectorExpiration ✖ ExtendSectorExpiration2 ✖ GetAvailableBalance ✔ GetBeneficiary ✔ GetOwner ✔ GetSectorSize ✔ GetVestingFunds ✔ IsControllingAddress ✔ OnDeferredCronEvent ✖ PreCommitSector ✖ PreCommitSectorBatch ✖ PreCommitSectorBatch2 ✖ ProveCommitAggregate ✖ ProveCommitSector ✖ ProveReplicaUpdates ✖ ProveReplicaUpdates2 ✖ Read fee debt ✖ Read initial pledge total ✖ Read peer ID, multiaddr ✔ Read pre-commit deposit ✖ RepayDebt ✔ ReportConsensusFault ✖ SubmitWindowedPoSt ✖ TerminateSectors ✖ WithdrawBalance ✔

Multisig

Method Supported? AddSigner ✔ Approve ✔ Cancel ✔ ChangeNumApprovalsThreshold ✖ Constructor ✖ List signers and threshold ✖ LockBalance ✔ Propose ✔ RemoveSigner ✔ SwapSigner ✔ UniversalReceiverHook ✔

Storage market

Method Supported? ActivateDeals ✖ AddBalance ✔ ComputeDataCommitment ✖ Constructor ✖ CronTick ✖ GetBalance ✔ GetDealActivation ✔ GetDealClient ✔ GetDealClientCollateral ✔ GetDealDataCommitment ✔ GetDealEpochPrice ✔ GetDealLabel ✔ GetDealProvider ✔ GetDealProviderCollateral ✔ GetDealTerm ✔ GetDealVerified ✔ OnMinerSectorsTerminate ✖ PublishStorageDeals ✔ VerifyDealsForActivation ✖ WithdrawBalance ✔

Storage power

Method Supported? Compute pledge collateral for new sector ✖ Constructor ✖ CreateMiner ✔ CurrentTotalPower ✖ EnrollCronEvent ✖ Get miner count, consensus count ✔ Get miner's QA power ✖ Get network bytes committed? ✖ Get network epoch pledge collateral ✖ Get network epoch QA power ✖ Get network total pledge collateral? ✖ MinerRawPower ✔ NetworkRawPower ✔ OnEpochTickEnd ✖ SubmitPoRepForBulkVerify ✖ UpdateClaimedPower ✖ UpdatePledgeTotal ✖

Verified registry

Method Supported? AddVerifiedClient ✔ AddVerifier ✖ ClaimAllocations ✖ Constructor ✖ ExtendClaimTerms ✔ GetClaims ✔ List claims ✖ List/check verifiers ✖ List/get allocations ✖ RemoveExpiredAllocations ✔ RemoveExpiredClaims ✔ RemoveVerifiedClientDataCap ✖ RemoveVerifier ✖ UniversalReceiverHook ✔

Import filecoin.sol

Thefilecoin.sol library is embeddable into your smart contract, which means it does not need be present on chain first. Instead, you can just import the library and call the available methods. Thefilecoin.sol library can beadded vianpm ormanually imported into your contract. Thenpm -based import is simpler, and is recommended.

Import filecoin.sol with npm

1. Installyarn
2. if you don't have it installed.
3. Installfilecoin.sol
4. :
5.

```

Copy yarnadd@zondax/filecoin.sol

```

Until mid-2023, Zondax was the legacy maintainer of Filecoin.sol. Protocol Labs took over the project, and are in the process of moving NPM packages over to theprotocollabs NPM account.

Import filecoin.sol manually

1. Navigate to your smart contract project folder
2. :
3.

```

Copy cd my-project

```

1. Create a folder namedlibs
2. :
3.

```

Copy mkdir libs

```

1. Move into thelibs
2. directory:
3.

```

Copy cd libs

```

1. Copy the Zondax contracts with the methods you wish to call fromthe contracts folder
2. intolibs
3. .
4.

Call a built-in actor

Once you've either imported particular contracts manually or simply installedfilecoin.sol using npm, create a callable method to access the built-in actor methods the way you normally would in a Solidity smart contract. Working examples of smart contracts that call built-in actor methods are available in thereference guide .

Previous Solidity libraries Next Filecoin.sol

Last updated5 months ago