State

The State methods are used to query, inspect, and interact with chain state. Most methods take a TipSetKey as a parameter. The state looked up is the parent state of the tipset. A nil TipSetKey can be provided as a param, this will cause the heaviest tipset in the chain to be used.

StateAccountKey

StateAccountKey returns the public key address of the given ID address for secp and bls accounts

Perms: read

Inputs:

```

Copy [ "f01234", [ { "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, { "/":"bafy2bzacebp3shtrn43k7g3unredz7fxn4gj533d3o43tqn2p2ipxxhrvchve" } ] ]

```

Response:"f01234"

StateActorCodeCIDs

StateActorCodeCIDs returns the CIDs of all the builtin actors for the given network version

Perms: read

Inputs:

```

Copy [21]

```

Response:{}

StateActorManifestCID

StateActorManifestCID returns the CID of the builtin actors manifest for the given network version

Perms: read

Inputs:

```

Copy [21]

```

Response:

```

Copy { "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }

```

StateAllMinerFaults

StateAllMinerFaults returns all non-expired Faults that occur within lookback epochs of the given tipset

Perms: read

Inputs:

```

Copy [ 10101, [ { "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, {

"/":"bafy2bzacebp3shtrn43k7g3unredz7fxn4gj533d3o43tqn2p2ipxxhrvchve" } ] ]

```

Response:

```

Copy [ { "Miner":"f01234", "Epoch":10101 } ]

```

StateCall

StateCall runs the given message and returns its result without any persisted changes.

StateCall applies the message to the tipset's parent state. The message is not applied on-top-of the messages in the passed-in tipset.

Perms: read

Inputs:

```

Copy [ { "Version":42, "To":"f01234", "From":"f01234", "Nonce":42, "Value":"0", "GasLimit":9, "GasFeeCap":"0", "GasPremium":"0", "Method":1, "Params":"Ynl0ZSBhcnJheQ==", "CID":{ "/":"bafy2bzacebbpdegvr3i4cosewthysg5xkxpqfn2wfcz6mv2hmoktwbdxkax4s" } }, [ { "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, { "/":"bafy2bzacebp3shtrn43k7g3unredz7fxn4gj533d3o43tqn2p2ipxxhrvchve" } ] ]

```

Response:

```

Copy { "MsgCid":{ "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, "Msg":{ "Version":42, "To":"f01234", "From":"f01234", "Nonce":42, "Value":"0", "GasLimit":9, "GasFeeCap":"0", "GasPremium":"0", "Method":1, "Params":"Ynl0ZSBhcnJheQ==", "CID":{ "/":"bafy2bzacebbpdegvr3i4cosewthysg5xkxpqfn2wfcz6mv2hmoktwbdxkax4s" } }, "MsgRct":{ "ExitCode":0, "Return":"Ynl0ZSBhcnJheQ==", "GasUsed":9, "EventsRoot":{ "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" } }, "GasCost":{ "Message":{ "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, "GasUsed":"0", "BaseFeeBurn":"0", "OverEstimationBurn":"0", "MinerPenalty":"0", "MinerTip":"0", "Refund":"0", "TotalCost":"0" }, "ExecutionTrace":{ "Msg":{ "From":"f01234", "To":"f01234", "Value":"0", "Method":1, "Params":"Ynl0ZSBhcnJheQ==", "ParamsCodec":42, "GasLimit":42, "ReadOnly":true, "CodeCid":{ "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" } }, "MsgRct":{ "ExitCode":0, "Return":"Ynl0ZSBhcnJheQ==", "ReturnCodec":42 }, "GasCharges":[ { "Name":"string value", "tg":9, "cg":9, "sg":9, "tt":60000000000 } ], "Subcalls":[ { "Msg":{ "From":"f01234", "To":"f01234", "Value":"0", "Method":1, "Params":"Ynl0ZSBhcnJheQ==", "ParamsCodec":42, "GasLimit":42, "ReadOnly":true, "CodeCid":{ "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" } }, "MsgRct":{ "ExitCode":0, "Return":"Ynl0ZSBhcnJheQ==", "ReturnCodec":42 }, "GasCharges":[ { "Name":"string value", "tg":9, "cg":9, "sg":9, "tt":60000000000 } ], "Subcalls":null } ] }, "Error":"string value", "Duration":60000000000 }

```

StateChangedActors

StateChangedActors returns all the actors whose states change between the two given state CIDs TODO: Should this take tipset keys instead?

Perms: read

Inputs:

```

Copy [ { "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, { "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" } ]

```

Response:

```
```

Copy { "t01236":{ "Code":{ "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, "Head":{
"/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, "Nonce":42, "Balance":"0",
"Address":"f01234" } }

```
```

StateCirculatingSupply

StateCirculatingSupply returns the exact circulating supply of Filecoin at the given tipset. This is not used anywhere in the protocol itself, and is only for external consumption.

Perms: read

Inputs:

```
```

Copy [[{ "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, {
"/":"bafy2bzacebp3shtrn43k7g3unredz7fxn4gj533d3o43tqn2p2ipxxhrvchve" }]]

```
```

Response:"0"

StateCompute

StateCompute is a flexible command that applies the given messages on the given tipset. The messages are run as though the VM were at the provided height.

When called, StateCompute will:

- Load the provided tipset, or use the current chain head if not provided
- Compute the tipset state of the provided tipset on top of the parent state
-
  - (note that this step runs before vmheight is applied to the execution)
-
  - Execute state upgrade if any were scheduled at the epoch, or in null blocks preceding the tipset
-
  - Call the cron actor on null blocks preceding the tipset
-
  - For each block in the tipset
-
  -
    - Apply messages in blocks in the specified
-
  -
    - Award block reward by calling the reward actor

-
  - Call the cron actor for the current epoch
- *
- If the specified vmheight is higher than the current epoch, apply any needed state upgrades to the state
- Apply the specified messages to the state
-

The vmheight parameter sets VM execution epoch, and can be used to simulate message execution in different network versions. If the specified vmheight epoch is higher than the epoch of the specified tipset, any state upgrades until the vmheight will be executed on the state before applying messages specified by the user.

Note that the initial tipset state computation is not affected by the vmheight parameter - only the messages in theapply set are

If the caller wants to simply compute the state, vmheight should be set to the epoch of the specified tipset.

Messages in theapply parameter must have the correct nonces, and gas values set.

Perms: read

Inputs:

```

Copy [ 10101, [ { "Version":42, "To":"f01234", "From":"f01234", "Nonce":42, "Value":"0", "GasLimit":9, "GasFeeCap":"0", "GasPremium":"0", "Method":1, "Params":"Ynl0ZSBhcnJheQ==", "CID":{ "/":"bafy2bzacebbpdegvr3i4cosewthysg5xkxpqfn2wfcz6mv2hmoktwbdxkax4s" } } ], [ { "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, { "/":"bafy2bzacebp3shtrn43k7g3unredz7fxn4gj533d3o43tqn2p2ipxxhrvchve" } ] ]

```

Response:

```

Copy { "Root":{ "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, "Trace":[ { "MsgCid":{ "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, "Msg":{ "Version":42, "To":"f01234", "From":"f01234", "Nonce":42, "Value":"0", "GasLimit":9, "GasFeeCap":"0", "GasPremium":"0", "Method":1, "Params":"Ynl0ZSBhcnJheQ==", "CID":{ "/":"bafy2bzacebbpdegvr3i4cosewthysg5xkxpqfn2wfcz6mv2hmoktwbdxkax4s" } }, "MsgRct":{ "ExitCode":0, "Return":"Ynl0ZSBhcnJheQ==", "GasUsed":9, "EventsRoot":{ "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" } }, "GasCost":{ "Message":{ "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, "GasUsed":"0", "BaseFeeBurn":"0", "OverEstimationBurn":"0", "MinerPenalty":"0", "MinerTip":"0", "Refund":"0", "TotalCost":"0" }, "ExecutionTrace":{ "Msg":{ "From":"f01234", "To":"f01234", "Value":"0", "Method":1, "Params":"Ynl0ZSBhcnJheQ==", "ParamsCodec":42, "GasLimit":42, "ReadOnly":true, "CodeCid":{ "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" } }, "MsgRct":{ "ExitCode":0, "Return":"Ynl0ZSBhcnJheQ==", "ReturnCodec":42 }, "GasCharges":[ { "Name":"string value", "tg":9, "cg":9, "sg":9, "tt":60000000000 } ], "Subcalls":[ { "Msg":{ "From":"f01234", "To":"f01234", "Value":"0", "Method":1, "Params":"Ynl0ZSBhcnJheQ==", "ParamsCodec":42, "GasLimit":42, "ReadOnly":true, "CodeCid":{ "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" } }, "MsgRct":{ "ExitCode":0, "Return":"Ynl0ZSBhcnJheQ==", "ReturnCodec":42 }, "GasCharges":[ { "Name":"string value", "tg":9, "cg":9, "sg":9, "tt":60000000000 } ], "Subcalls":null } ] }, "Error":"string value", "Duration":60000000000 } ] }

```

StateComputeDataCID

StateComputeDataCID computes DataCID from a set of on-chain deals

Perms: read

Inputs:

```

Copy [ "f01234", 8, [5432], [ { "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, { "/":"bafy2bzacebp3shtrn43k7g3unredz7fxn4gj533d3o43tqn2p2ipxxhrvchve" } ] ]

```

Response:

```

Copy { "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }

```

StateDealProviderCollateralBounds

StateDealProviderCollateralBounds returns the min and max collateral a storage provider can issue. It takes the deal size and verified status as parameters.

Perms: read

Inputs:

```

Copy [ 1032, true, [ { "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, { "/":"bafy2bzacebp3shtrn43k7g3unredz7fxn4gj533d3o43tqn2p2ipxxhrvchve" } ] ]

```

```
```

Response:

```
```

Copy { "Min":"0", "Max":"0" }

```
```

StateDecodeParams

StateDecodeParams attempts to decode the provided params, based on the recipient actor address and method number.

Perms: read

Inputs:

```
```

Copy ["f01234", 1, "Ynl0ZSBhcnJheQ==", [{ "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, { "/":"bafy2bzacebp3shtrn43k7g3unredz7fxn4gj533d3o43tqn2p2ipxxhrvchve" }]]

```
```

Response:{}

StateEncodeParams

StateEncodeParams attempts to encode the provided json params to the binary from

Perms: read

Inputs:

```
```

Copy [{ "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, 1, "json raw message"]

```
```

Response:"Ynl0ZSBhcnJheQ=="

StateGetActor

StateGetActor returns the indicated actor's nonce and balance.

Perms: read

Inputs:

```
```

Copy ["f01234", [{ "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, { "/":"bafy2bzacebp3shtrn43k7g3unredz7fxn4gj533d3o43tqn2p2ipxxhrvchve" }]]

```
```

Response:

```
```

Copy { "Code":{ "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, "Head":{ "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, "Nonce":42, "Balance":"0", "Address":"f01234" }

```
```

StateGetAllocation

StateGetAllocation returns the allocation for a given address and allocation ID.

Perms: read

Inputs:

```
Copy [ "f01234", 0, [ { "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, {
"/":"bafy2bzacebp3shtrn43k7g3unredz7fxn4gj533d3o43tqn2p2ipxxhrvchve" } ] ]
```

Response:

```
Copy { "Client":1000, "Provider":1000, "Data":{
"/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, "Size":1032, "TermMin":10101,
"TermMax":10101, "Expiration":10101 }
```

StateGetAllocationForPendingDeal

StateGetAllocationForPendingDeal returns the allocation for a given deal ID of a pending deal. Returns nil if pending allocation is not found.

Perms: read

Inputs:

```
Copy [ 5432, [ { "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, {
"/":"bafy2bzacebp3shtrn43k7g3unredz7fxn4gj533d3o43tqn2p2ipxxhrvchve" } ] ]
```

Response:

```
Copy { "Client":1000, "Provider":1000, "Data":{
"/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, "Size":1032, "TermMin":10101,
"TermMax":10101, "Expiration":10101 }
```

StateGetAllocations

StateGetAllocations returns the all the allocations for a given client.

Perms: read

Inputs:

```
Copy [ "f01234", [ { "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, {
"/":"bafy2bzacebp3shtrn43k7g3unredz7fxn4gj533d3o43tqn2p2ipxxhrvchve" } ] ]
```

Response:{}

StateGetBeaconEntry

StateGetBeaconEntry returns the beacon entry for the given filecoin epoch. If the entry has not yet been produced, the call will block until the entry becomes available

Perms: read

Inputs:

```

Copy [10101]

```

Response:

```

Copy { "Round":42, "Data":"Ynl0ZSBhcnJheQ==" }

```

StateGetClaim

StateGetClaim returns the claim for a given address and claim ID.

Perms: read

Inputs:

```

Copy [ "f01234", 0, [ { "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, { "/":"bafy2bzacebp3shtrn43k7g3unredz7fxn4gj533d3o43tqn2p2ipxxhrvchve" } ] ]

```

Response:

```

Copy { "Provider":1000, "Client":1000, "Data":{ "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, "Size":1032, "TermMin":10101, "TermMax":10101, "TermStart":10101, "Sector":9 }

```

StateGetClaims

StateGetClaims returns the all the claims for a given provider.

Perms: read

Inputs:

```

Copy [ "f01234", [ { "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, { "/":"bafy2bzacebp3shtrn43k7g3unredz7fxn4gj533d3o43tqn2p2ipxxhrvchve" } ] ]

```

Response:{}

StateGetNetworkParams

StateGetNetworkParams return current network params

Perms: read

Inputs:null

Response:

```

Copy { "NetworkName":"lotus", "BlockDelaySecs":42, "ConsensusMinerMinPower":"0", "SupportedProofTypes":[8], "PreCommitChallengeDelay":10101, "ForkUpgradeParams":{ "UpgradeSmokeHeight":10101, "UpgradeBreezeHeight":10101, "UpgradeIgnitionHeight":10101, "UpgradeLiftoffHeight":10101, "UpgradeAssemblyHeight":10101, "UpgradeRefuelHeight":10101, "UpgradeTapeHeight":10101, "UpgradeKumquatHeight":10101, "BreezeGasTampingDuration":10101, "UpgradeCalicoHeight":10101, "UpgradePersianHeight":10101, "UpgradeOrangeHeight":10101, "UpgradeClausHeight":10101, "UpgradeTrustHeight":10101, "UpgradeNorwegianHeight":10101, "UpgradeTurboHeight":10101,

"UpgradeHyperdriveHeight":10101, "UpgradeChocolateHeight":10101, "UpgradeOhSnapHeight":10101, "UpgradeSkyrHeight":10101, "UpgradeSharkHeight":10101, "UpgradeHyggeHeight":10101, "UpgradeLightningHeight":10101, "UpgradeThunderHeight":10101, "UpgradeWatermelonHeight":10101 }, "Eip155ChainID":123 }
```

StateGetRandomnessDigestFromBeacon

StateGetRandomnessDigestFromBeacon is used to sample the beacon for randomness.

Perms: read

Inputs:

```

Copy [ 10101, [ { "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, { "/":"bafy2bzacebp3shtrn43k7g3unredz7fxn4gj533d3o43tqn2p2ipxxhrvchve" } ] ]

```

Response:"Bw=="

StateGetRandomnessDigestFromTickets

StateGetRandomnessDigestFromTickets. is used to sample the chain for randomness.

Perms: read

Inputs:

```

Copy [ 10101, [ { "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, { "/":"bafy2bzacebp3shtrn43k7g3unredz7fxn4gj533d3o43tqn2p2ipxxhrvchve" } ] ]

```

Response:"Bw=="

StateGetRandomnessFromBeacon

StateGetRandomnessFromBeacon is used to sample the beacon for randomness.

Perms: read

Inputs:

```

Copy [ 2, 10101, "Ynl0ZSBhcnJheQ==", [ { "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, { "/":"bafy2bzacebp3shtrn43k7g3unredz7fxn4gj533d3o43tqn2p2ipxxhrvchve" } ] ]

```

Response:"Bw=="

StateGetRandomnessFromTickets

StateGetRandomnessFromTickets is used to sample the chain for randomness.

Perms: read

Inputs:

```

Copy [ 2, 10101, "Ynl0ZSBhcnJheQ==", [ { "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, { "/":"bafy2bzacebp3shtrn43k7g3unredz7fxn4gj533d3o43tqn2p2ipxxhrvchve" } ] ]

```

Response:"Bw=="

StateListActors

StateListActors returns the addresses of every actor in the state

Perms: read

Inputs:

```
```

Copy [[{ "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, { "/":"bafy2bzacebp3shtrn43k7g3unredz7fxn4gj533d3o43tqn2p2ipxxhrvchve" }]]

```
```

Response:

```
```

Copy ["f01234"]

```
```

StateListMessages

StateListMessages looks back and returns all messages with a matching to or from address, stopping at the given height.

Perms: read

Inputs:

```
```

Copy [{ "To":"f01234", "From":"f01234" }, [{ "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, { "/":"bafy2bzacebp3shtrn43k7g3unredz7fxn4gj533d3o43tqn2p2ipxxhrvchve" }], 10101]

```
```

Response:

```
```

Copy [{ "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }]

```
```

StateListMiners

StateListMiners returns the addresses of every miner that has claimed power in the Power Actor

Perms: read

Inputs:

```
```

Copy [[{ "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, { "/":"bafy2bzacebp3shtrn43k7g3unredz7fxn4gj533d3o43tqn2p2ipxxhrvchve" }]]

```
```

Response:

```
```

Copy ["f01234"]

```
```

StateLookupID

StateLookupID retrieves the ID address of the given address

Perms: read

Inputs:

```
```

Copy ["f01234", [{ "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, { "/":"bafy2bzacebp3shtrn43k7g3unredz7fxn4gj533d3o43tqn2p2ipxxhrvchve" }]]

```
```

Response:"f01234"

StateLookupRobustAddress

StateLookupRobustAddress returns the public key address of the given ID address for non-account addresses (multisig, miners etc)

Perms: read

Inputs:

```
```

Copy ["f01234", [{ "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, { "/":"bafy2bzacebp3shtrn43k7g3unredz7fxn4gj533d3o43tqn2p2ipxxhrvchve" }]]

```
```

Response:"f01234"

StateMarketBalance

StateMarketBalance looks up the Escrow and Locked balances of the given address in the Storage Market

Perms: read

Inputs:

```
```

Copy ["f01234", [{ "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, { "/":"bafy2bzacebp3shtrn43k7g3unredz7fxn4gj533d3o43tqn2p2ipxxhrvchve" }]]

```
```

Response:

```
```

Copy { "Escrow":"0", "Locked":"0" }

```
```

StateMarketDeals

StateMarketDeals returns information about every deal in the Storage Market

Perms: read

Inputs:

```
```

Copy [[{ "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, { "/":"bafy2bzacebp3shtrn43k7g3unredz7fxn4gj533d3o43tqn2p2ipxxhrvchve" }]]

```
```

Response:

```
Copy { "t026363":{ "Proposal":{ "PieceCID":{ "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" },
"PieceSize":1032, "VerifiedDeal":true, "Client":"f01234", "Provider":"f01234", "Label":"", "StartEpoch":10101,
"EndEpoch":10101, "StoragePricePerEpoch":"0", "ProviderCollateral":"0", "ClientCollateral":"0" }, "State":{
"SectorStartEpoch":10101, "LastUpdatedEpoch":10101, "SlashEpoch":10101, "VerifiedClaim":0 } } }
```

StateMarketParticipants

StateMarketParticipants returns the Escrow and Locked balances of every participant in the Storage Market

Perms: read

Inputs:

```

Copy [ [ { "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, {
"/":"bafy2bzacebp3shtrn43k7g3unredz7fxn4gj533d3o43tqn2p2ipxxhrvchve" } ] ]
```

Response:

```

Copy { "t026363":{ "Escrow":"0", "Locked":"0" } }
```

StateMarketStorageDeal

StateMarketStorageDeal returns information about the indicated deal

Perms: read

Inputs:

```

Copy [ 5432, [ { "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, {
"/":"bafy2bzacebp3shtrn43k7g3unredz7fxn4gj533d3o43tqn2p2ipxxhrvchve" } ] ]
```

Response:

```

Copy { "Proposal":{ "PieceCID":{ "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" },
"PieceSize":1032, "VerifiedDeal":true, "Client":"f01234", "Provider":"f01234", "Label":"", "StartEpoch":10101,
"EndEpoch":10101, "StoragePricePerEpoch":"0", "ProviderCollateral":"0", "ClientCollateral":"0" }, "State":{
"SectorStartEpoch":10101, "LastUpdatedEpoch":10101, "SlashEpoch":10101, "VerifiedClaim":0 } }
```

StateMinerActiveSectors

StateMinerActiveSectors returns info about sectors that a given miner is actively proving.

Perms: read

Inputs:

```

Copy [ "f01234", [ { "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, {
"/":"bafy2bzacebp3shtrn43k7g3unredz7fxn4gj533d3o43tqn2p2ipxxhrvchve" } ] ]
```

Response:

```
Copy [ { "SectorNumber":9, "SealProof":8, "SealedCID":{
"/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, "DealIDs":[5432], "Activation":10101,
"Expiration":10101, "DealWeight":"0", "VerifiedDealWeight":"0", "InitialPledge":"0", "ExpectedDayReward":"0",
"ExpectedStoragePledge":"0", "ReplacedSectorAge":10101, "ReplacedDayReward":"0", "SectorKeyCID":{
"/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, "SimpleQAPower":true } ]
```

StateMinerAllocated

StateMinerAllocated returns a bitfield containing all sector numbers marked as allocated in miner state

Perms: read

Inputs:

```
Copy [ "f01234", [ { "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, {
"/":"bafy2bzacebp3shtrn43k7g3unredz7fxn4gj533d3o43tqn2p2ipxxhrvchve" } ] ]
```

Response:

```
Copy [0]
```

StateMinerAvailableBalance

StateMinerAvailableBalance returns the portion of a miner's balance that can be withdrawn or spent

Perms: read

Inputs:

```
Copy [ "f01234", [ { "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, {
"/":"bafy2bzacebp3shtrn43k7g3unredz7fxn4gj533d3o43tqn2p2ipxxhrvchve" } ] ]
```

Response:"0"

StateMinerDeadlines

StateMinerDeadlines returns all the proving deadlines for the given miner

Perms: read

Inputs:

```
Copy [ "f01234", [ { "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, {
"/":"bafy2bzacebp3shtrn43k7g3unredz7fxn4gj533d3o43tqn2p2ipxxhrvchve" } ] ]
```

Response:

```
Copy [ { "PostSubmissions":[5,1], "DisputableProofCount":42 } ]
```

```

## StateMinerFaults

StateMinerFaults returns a bitfield indicating the faulty sectors of the given miner

Perms: read

Inputs:

```

Copy ["f01234", [{ "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, { "/":"bafy2bzacebp3shtrn43k7g3unredz7fxn4gj533d3o43tqn2p2ipxxhrvchve" }]]

```

Response:

```

Copy [5,1]

```

## StateMinerInfo

StateMinerInfo returns info about the indicated miner

Perms: read

Inputs:

```

Copy ["f01234", [{ "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, { "/":"bafy2bzacebp3shtrn43k7g3unredz7fxn4gj533d3o43tqn2p2ipxxhrvchve" }]]

```

Response:

```

Copy { "Owner":"f01234", "Worker":"f01234", "NewWorker":"f01234", "ControlAddresses":["f01234"], "WorkerChangeEpoch":10101, "PeerId":"12D3KooWGzxzKZYveHXtpG6AsrUJBcWxHBFS2HsEoGTxrMLvKXtf", "Multiaddrs":["Ynl0ZSBhcnJheQ=="], "WindowPoStProofType":8, "SectorSize":34359738368, "WindowPoStPartitionSectors":42, "ConsensusFaultElapsed":10101, "PendingOwnerAddress":"f01234", "Beneficiary":"f01234", "BeneficiaryTerm":{ "Quota":"0", "UsedQuota":"0", "Expiration":10101 }, "PendingBeneficiaryTerm":{ "NewBeneficiary":"f01234", "NewQuota":"0", "NewExpiration":10101, "ApprovedByBeneficiary":true, "ApprovedByNominee":true } }

```

## StateMinerInitialPledgeCollateral

StateMinerInitialPledgeCollateral returns the initial pledge collateral for the specified miner's sector

Perms: read

Inputs:

```

Copy ["f01234", { "SealProof":8, "SectorNumber":9, "SealedCID":{ "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, "SealRandEpoch":10101, "DealIDs":[5432], "Expiration":10101, "UnsealedCid":{ "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" } }, [{ "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, { "/":"bafy2bzacebp3shtrn43k7g3unredz7fxn4gj533d3o43tqn2p2ipxxhrvchve" }]]

```

Response:"0"

## StateMinerPartitions

StateMinerPartitions returns all partitions in the specified deadline

Perms: read

Inputs:

```
Copy ["f01234", 42, [{ "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, {
"/":"bafy2bzacebp3shtrn43k7g3unredz7fxn4gj533d3o43tqn2p2ipxxhrvchve" }]]
```

Response:

```
Copy [ { "AllSectors":[5,1], "FaultySectors":[5,1], "RecoveringSectors":[5,1], "LiveSectors":[5,1], "ActiveSectors":[5,1] } ]
```

StateMinerPower

StateMinerPower returns the power of the indicated miner

Perms: read

Inputs:

```
Copy [ "f01234", [ { "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, {
"/":"bafy2bzacebp3shtrn43k7g3unredz7fxn4gj533d3o43tqn2p2ipxxhrvchve" } ] ]
```

Response:

```
Copy { "MinerPower":{ "RawBytePower":"0", "QualityAdjPower":"0" }, "TotalPower":{ "RawBytePower":"0",
"QualityAdjPower":"0" }, "HasMinPower":true }
```

StateMinerPreCommitDepositForPower

StateMinerInitialPledgeCollateral returns the precommit deposit for the specified miner's sector

Perms: read

Inputs:

```
Copy [ "f01234", { "SealProof":8, "SectorNumber":9, "SealedCID":{
"/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, "SealRandEpoch":10101, "DealIDs":[5432],
"Expiration":10101, "UnsealedCid":{ "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" } }, [ {
"/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, {
"/":"bafy2bzacebp3shtrn43k7g3unredz7fxn4gj533d3o43tqn2p2ipxxhrvchve" } ] ]
```

Response:"0"

StateMinerProvingDeadline

StateMinerProvingDeadline calculates the deadline at some epoch for a proving period and returns the deadline-related calculations.

Perms: read

Inputs:

```
Copy [ "f01234", [ { "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, {
"/":"bafy2bzacebp3shtrn43k7g3unredz7fxn4gj533d3o43tqn2p2ipxxhrvchve" } ] ]
```

Response:

```
Copy { "CurrentEpoch":10101, "PeriodStart":10101, "Index":42, "Open":10101, "Close":10101, "Challenge":10101,
"FaultCutoff":10101, "WPoStPeriodDeadlines":42, "WPoStProvingPeriod":10101, "WPoStChallengeWindow":10101,
"WPoStChallengeLookback":10101, "FaultDeclarationCutoff":10101 }
```

StateMinerRecoveries

StateMinerRecoveries returns a bitfield indicating the recovering sectors of the given miner

Perms: read

Inputs:

```
Copy [ "f01234", [ { "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, {
"/":"bafy2bzacebp3shtrn43k7g3unredz7fxn4gj533d3o43tqn2p2ipxxhrvchve" } ] ]
```

Response:

```
Copy [5,1]
```

StateMinerSectorAllocated

StateMinerSectorAllocated checks if a sector number is marked as allocated.

Perms: read

Inputs:

```
Copy [ "f01234", 9, [ { "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, {
"/":"bafy2bzacebp3shtrn43k7g3unredz7fxn4gj533d3o43tqn2p2ipxxhrvchve" } ] ]
```

Response:true

StateMinerSectorCount

StateMinerSectorCount returns the number of sectors in a miner's sector set and proving set

Perms: read

Inputs:

```
Copy [ "f01234", [ { "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, {
"/":"bafy2bzacebp3shtrn43k7g3unredz7fxn4gj533d3o43tqn2p2ipxxhrvchve" } ] ]
```

Response:

```
Copy { "Live":42, "Active":42, "Faulty":42 }
```

StateMinerSectors

StateMinerSectors returns info about the given miner's sectors. If the filter bitfield is nil, all sectors are included.

Perms: read

Inputs:

```
Copy [ "f01234", [0], [ { "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, { "/":"bafy2bzacebp3shtrn43k7g3unredz7fxn4gj533d3o43tqn2p2ipxxhrvchve" } ] ]
```

Response:

```
Copy [ { "SectorNumber":9, "SealProof":8, "SealedCID":{ "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, "DealIDs":[5432], "Activation":10101, "Expiration":10101, "DealWeight":"0", "VerifiedDealWeight":"0", "InitialPledge":"0", "ExpectedDayReward":"0", "ExpectedStoragePledge":"0", "ReplacedSectorAge":10101, "ReplacedDayReward":"0", "SectorKeyCID":{ "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, "SimpleQAPower":true } ]
```

StateNetworkName

StateNetworkName returns the name of the network the node is synced to

Perms: read

Inputs:null

Response:"lotus"

StateNetworkVersion

StateNetworkVersion returns the network version at the given tipset

Perms: read

Inputs:

```
Copy [ [ { "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, { "/":"bafy2bzacebp3shtrn43k7g3unredz7fxn4gj533d3o43tqn2p2ipxxhrvchve" } ] ]
```

Response:21

StateReadState

StateReadState returns the indicated actor's state.

Perms: read

Inputs:

```
Copy [ "f01234", [ { "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, {
```

```
"/":"bafy2bzacebp3shtrn43k7g3unredz7fxn4gj533d3o43tqn2p2ipxxhrvchve" } ] ]
```

Response:

```
Copy { "Balance":"0", "Code":{ "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, "State":{} }
```

StateReplay

StateReplay replays a given message, assuming it was included in a block in the specified tipset.

If a tipset key is provided, and a replacing message is not found on chain, the method will return an error saying that the message wasn't found

If no tipset key is provided, the appropriate tipset is looked up, and if the message was gas-repriced, the on-chain message will be replayed - in that case the returned InvocResult.MsgCid will not match the Cid param

If the caller wants to ensure that exactly the requested message was executed, they MUST check that InvocResult.MsgCid is equal to the provided Cid. Without this check both the requested and original message may appear as successfully executed on-chain, which may look like a double-spend.

A replacing message is a message with a different CID, any of Gas values, and different signature, but with all other parameters matching (source/destination, nonce, params, etc.)

Perms: read

Inputs:

```
Copy [ [ { "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, {
"/":"bafy2bzacebp3shtrn43k7g3unredz7fxn4gj533d3o43tqn2p2ipxxhrvchve" } ], {
"/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" } ]
```

Response:

```
Copy { "MsgCid":{ "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, "Msg":{ "Version":42,
"To":"f01234", "From":"f01234", "Nonce":42, "Value":"0", "GasLimit":9, "GasFeeCap":"0", "GasPremium":"0", "Method":1,
"Params":"Ynl0ZSBhcnJheQ==", "CID":{ "/":"bafy2bzacebbpdegvr3i4cosewthysg5xkxpqfn2wfcz6mv2hmoktwbdxkax4s" } },
"MsgRct":{ "ExitCode":0, "Return":"Ynl0ZSBhcnJheQ==", "GasUsed":9, "EventsRoot":{
"/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" } }, "GasCost":{ "Message":{
"/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, "GasUsed":"0", "BaseFeeBurn":"0",
"OverEstimationBurn":"0", "MinerPenalty":"0", "MinerTip":"0", "Refund":"0", "TotalCost":"0" }, "ExecutionTrace":{ "Msg":{
"From":"f01234", "To":"f01234", "Value":"0", "Method":1, "Params":"Ynl0ZSBhcnJheQ==", "ParamsCodec":42,
"GasLimit":42, "ReadOnly":true, "CodeCid":{ "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }
}, "MsgRct":{ "ExitCode":0, "Return":"Ynl0ZSBhcnJheQ==", "ReturnCodec":42 }, "GasCharges":[ { "Name":"string value",
"tg":9, "cg":9, "sg":9, "tt":60000000000 } ], "Subcalls":[ { "Msg":{ "From":"f01234", "To":"f01234", "Value":"0", "Method":1,
"Params":"Ynl0ZSBhcnJheQ==", "ParamsCodec":42, "GasLimit":42, "ReadOnly":true, "CodeCid":{
"/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" } }, "MsgRct":{ "ExitCode":0,
"Return":"Ynl0ZSBhcnJheQ==", "ReturnCodec":42 }, "GasCharges":[ { "Name":"string value", "tg":9, "cg":9, "sg":9,
"tt":60000000000 } ], "Subcalls":null } ] }, "Error":"string value", "Duration":60000000000 }
```

StateSearchMsg

StateSearchMsg looks back up to limit epochs in the chain for a message, and returns its receipt and the tipset where it was executed

NOTE: If a replacing message is found on chain, this method will return a MsgLookup for the replacing message - the MsgLookup.Message will be a different CID than the one provided in the 'cid' param, MsgLookup.Receipt will contain the result of the execution of the replacing message.

If the caller wants to ensure that exactly the requested message was executed, they must check that MsgLookup.Message is equal to the provided 'cid', or set theallowReplaced parameter to false. Without this check, and withallowReplaced set to true, both the requested and original message may appear as successfully executed on-chain, which may look like a double-spend.

A replacing message is a message with a different CID, any of Gas values, and different signature, but with all other parameters matching (source/destination, nonce, params, etc.)

Perms: read

Inputs:

```

```

Copy [[{ "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, { "/":"bafy2bzacebp3shtrn43k7g3unredz7fxn4gj533d3o43tqn2p2ipxxhrvchve" }], { "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, 10101, true]

```

```

Response:

```

```

Copy { "Message":{ "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, "Receipt":{ "ExitCode":0, "Return":"Ynl0ZSBhcnJheQ==", "GasUsed":9, "EventsRoot":{ "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" } }, "ReturnDec":{}, "TipSet":[{ "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, { "/":"bafy2bzacebp3shtrn43k7g3unredz7fxn4gj533d3o43tqn2p2ipxxhrvchve" }], "Height":10101 }

```

```

StateSectorExpiration

StateSectorExpiration returns epoch at which given sector will expire

Perms: read

Inputs:

```

```

Copy ["f01234", 9, [{ "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, { "/":"bafy2bzacebp3shtrn43k7g3unredz7fxn4gj533d3o43tqn2p2ipxxhrvchve" }]]

```

```

Response:

```

```

Copy { "OnTime":10101, "Early":10101 }

```

```

StateSectorGetInfo

StateSectorGetInfo returns the on-chain info for the specified miner's sector. Returns null in case the sector info isn't found NOTE: returned info.Expiration may not be accurate in some cases, use StateSectorExpiration to get accurate expiration epoch

Perms: read

Inputs:

```

```

Copy ["f01234", 9, [{ "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, { "/":"bafy2bzacebp3shtrn43k7g3unredz7fxn4gj533d3o43tqn2p2ipxxhrvchve" }]]

```

```

Response:

```
Copy { "SectorNumber":9, "SealProof":8, "SealedCID":{
"/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, "DealIDs":[5432], "Activation":10101,
"Expiration":10101, "DealWeight":"0", "VerifiedDealWeight":"0", "InitialPledge":"0", "ExpectedDayReward":"0",
"ExpectedStoragePledge":"0", "ReplacedSectorAge":10101, "ReplacedDayReward":"0", "SectorKeyCID":{
"/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, "SimpleQAPower":true }
```

StateSectorPartition

StateSectorPartition finds deadline/partition with the specified sector

Perms: read

Inputs:

```
Copy [ "f01234", 9, [ { "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, {
"/":"bafy2bzacebp3shtrn43k7g3unredz7fxn4gj533d3o43tqn2p2ipxxhrvchve" } ] ]
```

Response:

```
Copy { "Deadline":42, "Partition":42 }
```

StateSectorPreCommitInfo

StateSectorPreCommitInfo returns the PreCommit info for the specified miner's sector. Returns nil and no error if the sector isn't precommitted.

Note that the sector number may be allocated while PreCommitInfo is nil. This means that either allocated sector numbers were compacted, and the sector number was marked as allocated in order to reduce size of the allocated sectors bitfield, or that the sector was precommitted, but the precommit has expired.

Perms: read

Inputs:

```
Copy [ "f01234", 9, [ { "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, {
"/":"bafy2bzacebp3shtrn43k7g3unredz7fxn4gj533d3o43tqn2p2ipxxhrvchve" } ] ]
```

Response:

```
Copy { "Info":{ "SealProof":8, "SectorNumber":9, "SealedCID":{
"/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, "SealRandEpoch":10101, "DealIDs":[5432],
"Expiration":10101, "UnsealedCid":{ "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" },
"PreCommitDeposit":"0", "PreCommitEpoch":10101 }
```

StateVMCirculatingSupplyInternal

StateVMCirculatingSupplyInternal returns an approximation of the circulating supply of Filecoin at the given tipset. This is the value reported by the runtime interface to actors code.

Perms: read

Inputs:

```
```
Copy [[{ "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, { "/":"bafy2bzacebp3shtrn43k7g3unredz7fxn4gj533d3o43tqn2p2ipxxhrvchve" }]]
```

Response:

```
Copy { "FilVested":"0", "FilMined":"0", "FilBurnt":"0", "FilLocked":"0", "FilCirculating":"0", "FilReserveDisbursed":"0" }
```

StateVerifiedClientStatus

StateVerifiedClientStatus returns the data cap for the given address. Returns nil if there is no entry in the data cap table for the address.

Perms: read

Inputs:

```
Copy [ "f01234", [ { "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, { "/":"bafy2bzacebp3shtrn43k7g3unredz7fxn4gj533d3o43tqn2p2ipxxhrvchve" } ] ]
```

Response:"0"

StateVerifiedRegistryRootKey

StateVerifiedRegistryRootKey returns the address of the Verified Registry's root key

Perms: read

Inputs:

```
Copy [ [ { "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, { "/":"bafy2bzacebp3shtrn43k7g3unredz7fxn4gj533d3o43tqn2p2ipxxhrvchve" } ] ]
```

Response:"f01234"

StateVerifierStatus

StateVerifierStatus returns the data cap for the given address. Returns nil if there is no entry in the data cap table for the address.

Perms: read

Inputs:

```
Copy [ "f01234", [ { "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, { "/":"bafy2bzacebp3shtrn43k7g3unredz7fxn4gj533d3o43tqn2p2ipxxhrvchve" } ] ]
```

Response:"0"

StateWaitMsg

StateWaitMsg looks back up to limit epochs in the chain for a message. If not found, it blocks until the message arrives on chain, and gets to the indicated confidence depth.

NOTE: If a replacing message is found on chain, this method will return a MsgLookup for the replacing message - the MsgLookup.Message will be a different CID than the one provided in the 'cid' param, MsgLookup.Receipt will contain the result of the execution of the replacing message.

If the caller wants to ensure that exactly the requested message was executed, they must check that MsgLookup.Message is equal to the provided 'cid', or set theallowReplaced parameter to false. Without this check, and withallowReplaced set to true, both the requested and original message may appear as successfully executed on-chain, which may look like a double-spend.

A replacing message is a message with a different CID, any of Gas values, and different signature, but with all other parameters matching (source/destination, nonce, params, etc.)

Perms: read

Inputs:

```
```

Copy [{ "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, 42, 10101, true]

```
```

Response:

```
```

Copy { "Message":{ "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, "Receipt":{
"ExitCode":0, "Return":"Ynl0ZSBhcnJheQ==", "GasUsed":9, "EventsRoot":{
"/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" } }, "ReturnDec":{}, "TipSet":[{
"/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, {
"/":"bafy2bzacebp3shtrn43k7g3unredz7fxn4gj533d3o43tqn2p2ipxxhrvchve" }], "Height":10101 }

```
```

Last updated5 months ago