Chain

The Chain method group contains methods for interacting with the blockchain, but that do not require any form of state computation.

ChainBlockstoreInfo

ChainBlockstoreInfo returns some basic information about the blockstore

Perms: read

Inputs:null

Response:

```
```

Copy { "abc":123 }

```
```

ChainCheckBlockstore

ChainCheckBlockstore performs an (asynchronous) health check on the chain/state blockstore if supported by the underlying implementation.

Perms: admin

Inputs:null

Response:{}

ChainDeleteObj

ChainDeleteObj deletes node referenced by the given CID

Perms: admin

Inputs:

```
```

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

```
```

Response:{}

ChainExport

ChainExport returns a stream of bytes with CAR dump of chain data. The exported chain data includes the header chain from the given tipset back to genesis, the entire genesis state, and the most recent 'nroots' state trees. If oldmsgskip is set, messages from before the requested roots are also not included.

Perms: read

Inputs:

```
```

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

```
```

Response:"Ynl0ZSBhcnJheQ=="

ChainExportRangeInternal

ChainExportRangeInternal triggers the export of a chain CAR-snapshot directly to disk. It is similar to ChainExport, except, depending on options, the snapshot can include receipts, messages and stateroots for the length between the specified

head and tail, thus producing "archival-grade" snapshots that include all the on-chain data. The header chain is included back to genesis and these snapshots can be used to initialize Filecoin nodes.

Perms: admin

Inputs:

```
Copy [ [ { "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, {
"/":"bafy2bzacebp3shtrn43k7g3unredz7fxn4gj533d3o43tqn2p2ipxxhrvchve" } ], [ {
"/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, {
"/":"bafy2bzacebp3shtrn43k7g3unredz7fxn4gj533d3o43tqn2p2ipxxhrvchve" } ], { "WriteBufferSize":123, "NumWorkers":123,
"IncludeMessages":true, "IncludeReceipts":true, "IncludeStateRoots":true } ]
```

Response:{}

ChainGetBlock

ChainGetBlock returns the block specified by the given CID.

Perms: read

Inputs:

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

Response:

```
Copy { "Miner":"f01234", "Ticket":{ "VRFProof":"Ynl0ZSBhcnJheQ==" }, "ElectionProof":{ "WinCount":9,
"VRFProof":"Ynl0ZSBhcnJheQ==" }, "BeaconEntries":[ { "Round":42, "Data":"Ynl0ZSBhcnJheQ==" } ], "WinPoStProof":[ {
"PoStProof":8, "ProofBytes":"Ynl0ZSBhcnJheQ==" } ], "Parents":[ {
"/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" } ], "ParentWeight":"0", "Height":10101,
"ParentStateRoot":{ "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" },
"ParentMessageReceipts":{ "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, "Messages":{
"/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, "BLSAggregate":{ "Type":2,
"Data":"Ynl0ZSBhcnJheQ==" }, "Timestamp":42, "BlockSig":{ "Type":2, "Data":"Ynl0ZSBhcnJheQ==" }, "ForkSignaling":42,
"ParentBaseFee":"0" }
```

ChainGetBlockMessages

ChainGetBlockMessages returns messages stored in the specified block.

Note: If there are multiple blocks in a tipset, it's likely that some messages will be duplicated. It's also possible for blocks in a tipset to have different messages from the same sender at the same nonce. When that happens, only the first message (in a block with lowest ticket) will be considered for execution

NOTE: THIS METHOD SHOULD ONLY BE USED FOR GETTING MESSAGES IN A SPECIFIC BLOCK

DO NOT USE THIS METHOD TO GET MESSAGES INCLUDED IN A TIPSET Use ChainGetParentMessages, which will perform correct message deduplication

Perms: read

Inputs:

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

Response:

```
```

Copy { "BlsMessages":[{ "Version":42, "To":"f01234", "From":"f01234", "Nonce":42, "Value":"0", "GasLimit":9, "GasFeeCap":"0", "GasPremium":"0", "Method":1, "Params":"Ynl0ZSBhcnJheQ==", "CID":{ "/":"bafy2bzacebbpdegvr3i4cosewthysg5xkxpqfn2wfcz6mv2hmoktwbdxkax4s" } }], "SecpkMessages":[{ "Message":{ "Version":42, "To":"f01234", "From":"f01234", "Nonce":42, "Value":"0", "GasLimit":9, "GasFeeCap":"0", "GasPremium":"0", "Method":1, "Params":"Ynl0ZSBhcnJheQ==", "CID":{ "/":"bafy2bzacebbpdegvr3i4cosewthysg5xkxpqfn2wfcz6mv2hmoktwbdxkax4s" } }, "Signature":{ "Type":2, "Data":"Ynl0ZSBhcnJheQ==" }, "CID":{ "/":"bafy2bzacebbpdegvr3i4cosewthysg5xkxpqfn2wfcz6mv2hmoktwbdxkax4s" } }], "Cids":[{ "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }] }

```
```

ChainGetEvents

ChainGetEvents returns the events under an event AMT root CID.

Perms: read

Inputs:

```
```

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

```
```

Response:

```
```

Copy [{ "Emitter":1000, "Entries":[{ "Flags":7, "Key":"string value", "Codec":42, "Value":"Ynl0ZSBhcnJheQ==" }] }]

```
```

ChainGetGenesis

ChainGetGenesis returns the genesis tipset.

Perms: read

Inputs:null

Response:

```
```

Copy { "Cids":null, "Blocks":null, "Height":0 }

```
```

ChainGetMessage

ChainGetMessage reads a message referenced by the specified CID from the chain blockstore.

Perms: read

Inputs:

```
```

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

```
```

Response:

```
```

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

```
```

ChainGetMessagesInTipset

ChainGetMessagesInTipset returns message stores in current tipset

Perms: read

Inputs:

```
```

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

```
```

Response:

```
```

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

```
```

ChainGetNode

Perms: read

Inputs:

```
```

Copy ["string value"]

```
```

Response:

```
```

Copy { "Cid":{ "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" }, "Obj":{} }

```
```

ChainGetParentMessages

ChainGetParentMessages returns messages stored in parent tipset of the specified block.

Perms: read

Inputs:

```
```

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

```
```

Response:

```
```

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

```
```

ChainGetParentReceipts

ChainGetParentReceipts returns receipts for messages in parent tipset of the specified block. The receipts in the list returned is one-to-one with the messages returned by a call to ChainGetParentMessages with the same blockCid.

Perms: read

Inputs:

```
```

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

```
```

Response:

```
```

Copy [{ "ExitCode":0, "Return":"Ynl0ZSBhcnJheQ==", "GasUsed":9, "EventsRoot":{ "/":"bafy2bzacea3wsdh6y3a36tb3skempjoxqpuyompjbmfeyf34fi3uy6uue42v4" } }]

```
```

ChainGetPath

ChainGetPath returns a set of revert/apply operations needed to get from one tipset to another, for example:

```
```

Copy to ^ from tAA ^ ^ tBA tAB ^---*--^ ^ tRR

```
```

Would return[revert(tBA), apply(tAB), apply(tAA)]

Perms: read

Inputs:

```
```

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

```
```

Response:

```
```

Copy [{ "Type":"string value", "Val":{ "Cids":null, "Blocks":null, "Height":0 } }]

```
```

ChainGetTipSet

ChainGetTipSet returns the tipset specified by the given TipSetKey.

Perms: read

Inputs:

```
```

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

```
```

Response:

```
```

Copy { "Cids":null, "Blocks":null, "Height":0 }

```

ChainGetTipSetAfterHeight

ChainGetTipSetAfterHeight looks back for a tipset at the specified epoch. If there are no blocks at the specified epoch, the first non-nil tipset at a later epoch will be returned.

Perms: read

Inputs:

```

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

```

Response:

```

Copy { "Cids":null, "Blocks":null, "Height":0 }

```

ChainGetTipSetByHeight

ChainGetTipSetByHeight looks back for a tipset at the specified epoch. If there are no blocks at the specified epoch, a tipset at an earlier epoch will be returned.

Perms: read

Inputs:

```

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

```

Response:

```

Copy { "Cids":null, "Blocks":null, "Height":0 }

```

ChainHasObj

ChainHasObj checks if a given CID exists in the chain blockstore.

Perms: read

Inputs:

```

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

```

Response:true

ChainHead

ChainHead returns the current head of the chain.

Perms: read

Inputs:null

Response:

```
Copy { "Cids":null, "Blocks":null, "Height":0 }
```

ChainHotGC

ChainHotGC does online (badger) GC on the hot store; only supported if you are using the splitstore

Perms: admin

Inputs:

```
Copy [ { "Threshold":12.3, "Periodic":true, "Moving":true } ]
```

Response:{}

ChainNotify

ChainNotify returns channel with chain head updates. First message is guaranteed to be of len == 1, and type == 'current'.

Perms: read

Inputs:null

Response:

```
Copy [ { "Type":"string value", "Val":{ "Cids":null, "Blocks":null, "Height":0 } } ]
```

ChainPrune

ChainPrune forces compaction on cold store and garbage collects; only supported if you are using the splitstore

Perms: admin

Inputs:

```
Copy [ { "MovingGC":true, "RetainState":9 } ]
```

Response:{}

ChainPutObj

ChainPutObj puts a given object into the block store

Perms: admin

Inputs:

```
Copy [{}]
```

Response:{}

ChainReadObj

ChainReadObj reads ipld nodes referenced by the specified CID from chain blockstore and returns raw bytes.

Perms: read

Inputs:

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

Response:"Ynl0ZSBhcnJheQ=="

ChainSetHead

ChainSetHead forcefully sets current chain head. Use with caution.

Perms: admin

Inputs:

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

Response:{}

ChainStatObj

ChainStatObj returns statistics about the graph referenced by 'obj'. If 'base' is also specified, then the returned stat will be a diff between the two objects.

Perms: read

Inputs:

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

Response:

```
Copy { "Size":42, "Links":42 }
```

ChainTipSetWeight

ChainTipSetWeight computes weight for the specified tipset.

Perms: read

Inputs:

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

Response:"0"

Last updated 5 months ago

Last updated 5 months ago