

JSON-RPC API

There are several OP Mainnet components with an RPC API, which are reviewed in this guide.

⚠ Use [eth_gasPrice \(opens in a new tab\)](#) instead of `rollup_gasPrices` for the L2 gas price. For the L1 gas price, you can call the `GasPriceOracle.sl1BaseFee` function [\(opens in a new tab\)](#). If you want to estimate the cost of a transaction, you can [use the SDK](#).

op-node

op-node implements most rollup-specific functionality as Consensus-Layer, similar to a L1 beacon-node. The following RPC methods are broken down by namespace.

The following examples show you how to make requests with [curl \(opens in a new tab\)](#) and [cast \(opens in a new tab\)](#).

Protip: piping these commands in [tq \(opens in a new tab\)](#) will give you nicely formatted JSON responses.

```
cast rpc optimism_syncStatus --rpc-url http://localhost:9545 | jq
```

optimism

Optimism specific rollup methods.

optimism_outputAtBlock

Get the output root at a specific block. This method is documented in [the specifications \(opens in a new tab\)](#).

```
curl cast curl
```

```
-X
```

```
POST
```

```
-H
```

```
"Content-Type: application/json"
```

```
--data \'{"jsonrpc":"2.0","method":"optimism_outputAtBlock","params":[{},"id":1}\' \ http://localhost:9545 Sample success output:
```

```
{ "jsonrpc": "2.0", "id": 1, "result": [ "0x0000000000000000000000000000000000000000000000000000000000000000",  
"0xab711e34c1387c8c56d0def8ce77e454d6a0bfd26cef2396626202238442421" ] }
```

optimism_syncStatus

Get the synchronization status.

```
curl cast curl
```

```
-X
```

```
POST
```

```
-H
```

```
"Content-Type: application/json"
```

```
--data \'{"jsonrpc":"2.0","method":"optimism_syncStatus","params":[{},"id":1}\' \ http://localhost:9545 Sample success output:
```

```
{ "current_l1": { "hash":
```

```
"0xff3b3253058411b727ac662f4c9ae1698918179e02eeced304beb1a1ae8fc4fd", "number":
```

```
4427350, "parentHash":
```

```
"0xb26586390c3f04678706dde13abfb5c6e6bb545e59c22774e651db224b16cd48", "timestamp":
```

```
1696478784 }, "current_l1_finalized": { "hash":
```

```
"0x7157f91b8ae21ef869c604e5b268e392de5aa69a9f44466b9b0f838d56426541", "number":
```

```
4706784, "parentHash":
```

```
"0x1ac2612a500b9facd650950b8755d97cf2470818da2d88552dea7cd563e86a17", "timestamp":
```

```
1700160084 }, "head_l1": { "hash":
```

```
"0x6110a8e6ed4c4aaab20477a3eac81bf99e505bf6370cd4d2e3c6d34aa5f4059a", "number":
```

```
4706863, "parentHash":
```

```
"0xee8a9cba5d93481f1145c24890fd8f536384f3c3c043f40006650538fbdcb56", "timestamp":
```

```
1700161272 }, "safe_l1": { "hash":
```

```
"0x8407c9968ce278ab435eeaced18ba8f2f94670ad9d3bdd170560932cf46e2804", "number":
```

```
4706811, "parentHash":
```

```
"0x6593ccab3e77276418ff691f6e4e75597af18505373522480fdd97219c06ef", "timestamp":
```

```
1700160480 }, "finalized_l1": { "hash":
```

```
"0x7157f91b8ae21ef869c604e5b268e392de5aa69a9f44466b9b0f838d56426541", "number":
```

```
4706784, "parentHash":
```

```
"0x1ac2612a500b9facd650950b8755d97cf2470818da2d88552dea7cd563e86a17", "timestamp":
```

```
1700160084 }, "unsafe_l2": { "hash":
```

```
"0x9a3b2edab72150de252d45cabe2f1ac57d48ddd52bb891831fed00e89408fe4", "number":
```

```
2338094, "parentHash":
```

```
"0x935b94ec0bac0e63c67a870b1a97d79e3fa84dda86d31996516cb2f940753f53", "timestamp":
```

```
1696478728, "l1origin": { "hash":
```

```
"0x38731e0a6eeb40091f0c4a00650e911c57d054aaeb5b158f55cd5705fa6a3ebf", "number":
```

```
4427339 }, "sequenceNumber":
```

```
3 }, "safe_l2": { "hash":
```

```
"0x9a3b2edab72150de252d45cabe2f1ac57d48ddd52bb891831fed00e89408fe4", "number":
```

```
2338094, "parentHash":
```

```

"0x935b94ec0bac0e63c67a870b1a97d79e3fa84dda86d31996516cb2f940753f53", "timestamp":
1696478728, "l1origin": { "hash":
"0x38731e0a6eeb40091f0c4a00650e911c57d054aaeb5b158f55cd5705fa6a3ebf", "number":
4427339 }, "sequenceNumber":
3 }, "finalized_l2": { "hash":
"0x285b03afb46faad747be1ca7ab6ef5e0ff01fe04e4eeabafcf54f129d180fad2", "number":
2337942, "parentHash":
"0x7e7136cbe1fd1ccdcdaa81577a1732776a01c0108ab5f98986cf997724eb48ac", "timestamp":
1696478424, "l1origin": { "hash":
"0x983309dadf7e0ab8447f3050f2a85b179e9acde1cd884f883fb331908c356412", "number":
4427314 }, "sequenceNumber":
7 }, "pending_safe_l2": { "hash":
"0x9a3b2edab72150de252d45cabe2f1ac57d48ddd52bb891831ffed00e89408fe4", "number":
2338094, "parentHash":
"0x935b94ec0bac0e63c67a870b1a97d79e3fa84dda86d31996516cb2f940753f53", "timestamp":
1696478728, "l1origin": { "hash":
"0x38731e0a6eeb40091f0c4a00650e911c57d054aaeb5b158f55cd5705fa6a3ebf", "number":
4427339 }, "sequenceNumber":
3 }, "queued_unsafe_l2": { "hash":
"0x3af253f5b993f58ffdd5e594b3f53f5b7b254cdc18f4bdb13ea7331149942db", "number":
4054795, "parentHash":
"0x284b7dc92bac97be8ec3b2cf548e75208eb288704de381f2557938ecdf86539d", "timestamp":
1699912130, "l1origin": { "hash":
"0x1490a63c372090a0331e05e63ec6a7a6e84835f91776306531f28b4217394d76", "number":
4688196 }, "sequenceNumber":
2 }, "engine_sync_target": { "hash":
"0x9a3b2edab72150de252d45cabe2f1ac57d48ddd52bb891831ffed00e89408fe4", "number":
2338094, "parentHash":
"0x935b94ec0bac0e63c67a870b1a97d79e3fa84dda86d31996516cb2f940753f53", "timestamp":
1696478728, "l1origin": { "hash":
"0x38731e0a6eeb40091f0c4a00650e911c57d054aaeb5b158f55cd5705fa6a3ebf", "number":
4427339 }, "sequenceNumber":
3 } }

```

```
"0x16fc5058f25648194471939df75cf27a2fdc48bc" , "t1_system_config_address" :  
"0x034edd2a225f7f429a63e0f1d2084b9e0a93b538" , "protocol_versions_address" :  
"0x79add5713b383daa0a138d3c4780c7a1804a8090" }
```

optimism_version

Get the software version.

At the moment, building from source will not give you the correct version, but our docker images will.

```
curl cast curl  
-X  
POST  
-H  
"Content-Type: application/json"  
--data \ {"jsonrpc":"2.0","method":"optimism_version","params":[],"id":1} \ http://localhost:9545 Sample success output:  
{ "jsonrpc" : "2.0" , "id" : 1 , "result" : "v0.0.0-"
```

opp2p

Theopp2p namespace handles peer interactions.

opp2p_self

Returns your node's information.

```
curl cast curl  
-X  
POST  
-H  
"Content-Type: application/json"  
--data \ {"jsonrpc":"2.0","method":"opp2p_self","params":[],"id":1} \ http://localhost:9545 Sample success output:  
{ "jsonrpc" :  
"2.0" , "id" :  
1 , "result" : { "peerID" :  
"16Uiu2HAM2y6DXp6THWHCyquczNUh8gVAm4spo6hjP3Ns1dGRiAdE" , "nodeID" :  
"75a52a90fe5f972171fefce2399ca5a73191c654e7c7ddfd71edf4fca6697f0" , "userAgent" :  
"" , "protocolVersion" :  
"" , "ENR" :  
"enr:-J-4QFOtl_hDBa_kilrQcg4ITJt9VMAuDLCbgAAKMa--  
WfxoPml1xDYxypUG7lsWga83FOlvr78LG3oH8CfzRzUmsDyGAYvKqIZ2gmIkgnY0gmlwhGxAaceHb3BzdGFja4Xc76gFAIlzZWNwMjU2azGhAnAON-  
FvpiWY2iG_LXJDYosknGyikaajPDd1cQARsVnBg3RjclIkBoN1ZHCC0Vs" , "addresses" : [ "/ip4/127.0.0.1/tcp/9222/p2p/16Uiu2HAM2y6DXp6THWHCyquczNUh8gVAm4spo6hjP3Ns1dGRiAdE" ,  
"/ip4/192.168.1.71/tcp/9222/p2p/16Uiu2HAM2y6DXp6THWHCyquczNUh8gVAm4spo6hjP3Ns1dGRiAdE" ,  
"/ip4/108.64.105.199/tcp/9222/p2p/16Uiu2HAM2y6DXp6THWHCyquczNUh8gVAm4spo6hjP3Ns1dGRiAdE" ] , "protocols" :  
null , "connectedness" :  
0 , "direction" :  
0 , "protected" :  
false , "chainID" :  
0 , "latency" :  
0 , "gossipBlocks" :  
true , "scores" : { "gossip" : { "total" :  
0 , "blocks" : { "timeInMesh" :  
0 , "firstMessageDeliveries" :  
0 , "meshMessageDeliveries" :  
0 , "invalidMessageDeliveries" :  
0 } , "IPColocationFactor" :  
0 , "behavioralPenalty" :  
0 } , "reqResp" : { "validResponses" :  
0 , "errorResponses" :  
0 , "rejectedPayloads" :  
0 } } } }
```

opp2p_peers

Returns a list of your node's peers.

```
curl cast curl  
-X  
POST  
-H  
"Content-Type: application/json"  
--data \ {"jsonrpc":"2.0","method":"opp2p_peers","params":[true],"id":1} \ http://localhost:9545 Sample success output:  
{ "jsonrpc" :
```

```
"2.0", "id" :

1, "result" : { "totalConnected" :

20, "peers" : { "16Uiu2HAKvNYscHu4V1uj6fVWkwrAMCRsqXDSq4mUbhpGq4LttYsC" : { "peerID" :

"16Uiu2HAKvNYscHu4V1uj6fVWkwrAMCRsqXDSq4mUbhpGq4LttYsC", "nodeID" :

"d693c5b58424016c0c38ec5539c272c754cb6b8007b322e0ecf16a4ee13f96fb", "userAgent" :

"optimism", "protocolVersion" :

"", "ENR" :

"", "addresses" : [ "/ip4/20.249.62.215/tcp/9222/p2p/16Uiu2HAKvNYscHu4V1uj6fVWkwrAMCRsqXDSq4mUbhpGq4LttYsC" ], "protocols" : [ "/ipfs/ping/1.0.0", "/meshsub/1.0.0", "/meshsub/1.1.0",
"/opstack/req/payload_by_number/11155420/0", "/floodsub/1.0.0", "/ipfs/id/1.0.0", "/ipfs/id/push/1.0.0" ], "connectedness" :

1, "direction" :

1, "protected" :

false, "chainID" :

0, "latency" :

0, "gossipBlocks" :

true, "scores" : { "gossip" : { "total" :

-5.04, "blocks" : { "timelnMesh" :

0, "firstMessageDeliveries" :

0, "meshMessageDeliveries" :

0, "invalidMessageDeliveries" :

0 }, "IPColocationFactor" :

0, "behavioralPenalty" :

0 }, "reqResp" : { "validResponses" :

0, "errorResponses" :

0, "rejectedPayloads" :

0 } } }, ... "bannedPeers" : [], "bannedIPS" : [], "bannedSubnets" : [] } }
```

opp2p_peerStats

Returns your peer stats.

```
curl cast curl

-X

POST

-H

"Content-Type: application/json"

--data \{"jsonrpc":"2.0","method":"opp2p_peerStats","params":[],"id":1}\ http://localhost:9545 Sample success output:

{ "jsonrpc" :

"2.0", "id" :

1, "result" : { "connected" :

20, "table" :

94, "blocksTopic" :

20, "blocksTopicV2" :

18, "banned" :

0, "known" :

71 } }
```

opp2p_discoveryTable

Returns your peer discovery table.

```
curl cast curl

-X

POST

-H

"Content-Type: application/json"

--data \{"jsonrpc":"2.0","method":"opp2p_discoveryTable","params":[],"id":1}\ http://localhost:9545 Sample success output:

{ "jsonrpc" :

"2.0", "id" :

1, "result" : [ "enr:-J24QGC_SzoGG4EqyvO_082paQOwhvECeWGT-kaenrHdE2_iLTLeGmH-IOVpqEjC0L-yWmkl-
c7598VaCjHQRNwn1CyGAYsqDgrFgmikgnY0gmlwhNgp3WbHb3BzdGFja4OFQgCjC2VjcDI1NmsxoQPiq20PNZYzyvpEifcGvROXHfM94JeWSgDL07I2hSI0d4N0Y3CCJAaDdWRwgiQG", "enr:-
J24QKvt2ThBM8-
FFeHfAmpoaVLdFVD2cw1cRpNuwmvH_bQtQ1dqrZw9FqiMbXbFRQf9lvjrIKSFLOdbsRALIFATICGAYuQCJigmlkgnY0gmlwhKI3ZuaHb3BzdGFja4O6BACJc2VjcDI1NmsxoQLQRz2CH95qQd6vmF5saV-
W0oTZobNfSt-FudVa7R35nYN0Y3CClyuDdWRwgiMr", "enr:-J24QEMvLGMrbBdW48cHq9GzHSDiPIJ3eUpZiOMMXwvxN0JLQMZsS3-J6DKP2Uu--
65BC5IAQF3zv7ukQDbEjuZpr5iGAYs0a4segmlkgnY0gmlwhEEVRWmHb3BzdGFja4OFQgCjC2VjcDI1NmsxoQJI0VT2eysk6v3t63pOZIGgJReDnNqh30roStkUlrIPGCoN0Y3CCJAaDdWRwgiQG", ...
"enr:-J24QOdxnjbP7MkCFBuEjWe50NGdtAECj8tyrDBmXm8Y-JydVeoNF_IXNqodLUqeAfuAe3gcLceju96NVf8as78Eaq-
GAYlu7qN_gmlkgnY0gmlwhAUJaBCHb3BzdGFja4OFQgCjC2VjcDI1NmsxoQKtb20tInUf5v-jP7YXZXI1wcfesSuPvsVd2niINV3BxHBYN0Y3CCJAaDdWRwgiQG" ] }
```

opp2p_blockPeer

Blocks a peer.

```
curl cast curl
```

-X

POST

-H

"Content-Type: application/json"

--data \ {"jsonrpc":"2.0","method":"opp2p_blockPeer","params":[],"id":1}' \ http://localhost:9545 Sample success output:

```
{ "jsonrpc" :  
  "2.0" , "id" :  
  1 , "result" :  
  null }
```

opp2p_unblockPeer

Unblocks a peer.

curl cast curl

-X

POST

-H

"Content-Type: application/json"

--data \ {"jsonrpc":"2.0","method":"opp2p_unblockPeer","params":[],"id":1}' \ http://localhost:9545 Sample success output:

```
{ "jsonrpc" :  
  "2.0" , "id" :  
  1 , "result" :  
  null }
```

opp2p_listBlockedPeers

Returns a list of your blocked peers.

curl cast curl

-X

POST

-H

"Content-Type: application/json"

--data \ {"jsonrpc":"2.0","method":"opp2p_listBlockedPeers","params":[],"id":1}' \ http://localhost:9545 Sample success output:

```
{ "jsonrpc": "2.0", "id": 1, "result": [ "16Uiu2HAmV3PueiaHj7Rg2bs3mrRUo2RVhjXRMpH67k9iZquDGQ8v" ] }
```

opp2p_blockAddr

Blocks ip address.

curl cast curl

-X

POST

-H

"Content-Type: application/json"

--data \ {"jsonrpc":"2.0","method":"opp2p_blockAddr","params":[""],"id":1}' \ http://localhost:9545 Sample success output:

```
{ "jsonrpc" :  
  "2.0" , "id" :  
  1 , "result" :  
  null }
```

opp2p_unblockAddr

Unblocks ip address.

curl cast curl

-X

POST

-H

"Content-Type: application/json"

--data \ {"jsonrpc":"2.0","method":"opp2p_unblockAddr","params":[""],"id":1}' \ http://localhost:9545 Sample success output:

```
{ "jsonrpc" :  
  "2.0" , "id" :  
  1 , "result" :  
  null }
```

opp2p_listBlockedAddrs

Returns a list of blocked ip addresses.

curl cast curl

-X

POST

-H

"Content-Type: application/json"

--data \ {"jsonrpc":"2.0","method":"opp2p_listBlockedAddrs","params":[],"id":1}' \ http://localhost:9545 Sample success output:

```
{ "jsonrpc": "2.0", "id": 1, "result": [ "2607:f8b0:4002:c0c::65" ] }
```

opp2p_blockSubnet

Blocks subnet.

curl cast curl

-X

POST

-H

"Content-Type: application/json"

--data \ {"jsonrpc":"2.0","method":"opp2p_blockSubnet","params":[""],"id":1}' \ http://localhost:9545 Sample success output:

```
{ "jsonrpc" :
```

```
"2.0" , "id" :
```

```
1 , "result" :
```

```
null }
```

opp2p_unblockSubnet

Unblocks subnet.

curl cast curl

-X

POST

-H

"Content-Type: application/json"

--data \ {"jsonrpc":"2.0","method":"opp2p_unblockSubnet","params":[""],"id":1}' \ http://localhost:9545 Sample success output:

```
{ "jsonrpc" :
```

```
"2.0" , "id" :
```

```
1 , "result" :
```

```
null }
```

opp2p_listBlockedSubnets

Returns list of blocked subnets.

curl cast curl

-X

POST

-H

"Content-Type: application/json"

--data \ {"jsonrpc":"2.0","method":"opp2p_listBlockedSubnets","params":[],"id":1}' \ http://localhost:9545 Sample success output:

```
{ "jsonrpc" :
```

```
"2.0" , "id" :
```

```
1 , "result" : [ ] }
```

opp2p_protectPeer

Protect peer.

curl cast curl

-X

POST

-H

"Content-Type: application/json"

--data \ {"jsonrpc":"2.0","method":"opp2p_protectPeer","params":[""],"id":1}' \ http://localhost:9545 Sample success output:

```
{ "jsonrpc" :
```

```
"2.0" , "id" :
```

```
1 , "result" :
```

```
null }
```

opp2p_unprotectPeer

Unprotect peer.

curl cast curl

-X

POST

-H

"Content-Type: application/json"

--data \ {"jsonrpc":"2.0","method":"opp2p_unprotectPeer","params":[""],"id":1} \ http://localhost:9545 Sample success output:

```
{ "jsonrpc" :  
  "2.0" , "id" :  
  1 , "result" :  
  null }
```

opp2p_connectPeer

Connect peer.

curl cast curl

-X

POST

-H

"Content-Type: application/json"

--data \ {"jsonrpc":"2.0","method":"opp2p_connectPeer","params":[],"id":1} \ http://localhost:9545 Sample success output:

```
{ "jsonrpc" :  
  "2.0" , "id" :  
  1 , "result" :  
  null }
```

opp2p_disconnectPeer

Disconnect peer.

curl cast curl

-X

POST

-H

"Content-Type: application/json"

--data \ {"jsonrpc":"2.0","method":"opp2p_disconnectPeer","params":[""],"id":1} \ http://localhost:9545 Sample success output:

```
{ "jsonrpc" :  
  "2.0" , "id" :  
  1 , "result" :  
  null }
```

admin

todo: add admin namespace description

admin_resetDerivationPipeline

Resets the derivation pipeline.

curl cast curl

-X

POST

-H

"Content-Type: application/json"

--data \ {"jsonrpc":"2.0","method":"admin_resetDerivationPipeline","params":[],"id":1} \ http://localhost:9545 Sample success output:

todo: add Sample success output

admin_startSequencer

Starts thh sequencer.

curl cast curl

-X

POST

-H

"Content-Type: application/json"

--data \ {"jsonrpc":"2.0","method":"admin_startSequencer","params":[],"id":1} \ http://localhost:9545 Sample success output:

todo: add Sample success output

admin_stopSequencer

Stops sequencer.

curl cast curl

-X

POST

-H

"Content-Type: application/json"

--data \ {"jsonrpc":"2.0","method":"admin_stopSequencer","params":[],"id":1} \ http://localhost:9545 Sample success output:

todo: add Sample success output

admin_sequencerActive

TODO: add description

curl cast curl

-X

POST

-H

"Content-Type: application/json"

--data \ '{"jsonrpc":"2.0","method":"admin_sequencerActive","params":[todo],"id":1}' \ http://localhost:9545 Sample success output:

todo: add Sample success output

op-geth

op-geth implements the Execution-Layer, with minimal changes for a secure Ethereum-equivalent application environment.

The execution engine's RPC interface is identical to [the upstream Geth RPC interface\(opens in a new tab\)](#). The responses are nearly identical too, except we also include the L1 gas usage and price information.

[Network Upgrades Software Releases](#)