节点引擎服务（NES）

开发指南

文档版本 01

发布日期 2024-03-11

华为云计算技术有限公司

商标声明


和其他华为商标均为华为技术有限公司的商标。
本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为云计算技术有限公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为云计算技术有限公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

目 录

1 以太坊 Ethereum 节点引擎

1.1 以太坊 Ethereum 介绍

以太坊

以太坊是一条区块链，其中嵌入了计算机。 它是以去中心化、无需许可、抗审查的方式构建应用程序和组织的基础。权益证明 (POS) 是支撑以太坊**共识机制**的基础。 以太坊于 2022 年启动了权益证明机制，这是因为和原先的**工作量证明**架构相比，以太坊更安全、能耗更低并且更利于实现新的扩容解决方案。

以太坊官方连接：以太坊简介，权益证明机制(POS)，Github，以太坊官网

用户可以通过使用华为云公链节点引擎，来提升区块链使用与开发的效率，增强其稳定性与私密性。**华为云将永远不会收集用户的区块链地址。**

说明

- **支持网络**
  - 以太坊主网：HTTP/WSS
  - Goerli测试网：HTTP/WSS
  - Sepolia测试网：HTTP/WSS
- **执行层API支持清单**
- **共识层API支持清单**

1.2 JSON-RPC API 请求示例

1.2.1 使用 cURL 发送 JSON-RPC API 请求

1.2.1.1 执行层请求示例

Request example(With Credential):

```
curl -X POST https://your-http-endpoint/your-credential \
-H 'Content-Type: application/json' \
```

```
-d '{
"jsonrpc": "2.0",
"method":  "eth_blockNumber",
"params": [],
"id": 1
}'
```

Request example(With IAM Token):

```
curl -X POST -H 'X-Auth-Token:your-iam-token' https://your-http-endpoint \
-H 'Content-Type: application/json' \
-d '{
"jsonrpc": "2.0",
"method":  "eth_blockNumber",
"params": [],
"id": 1
}'
```

Response example:

```
{
"jsonrpc": "2.0",
"id": 1,
"result": "00f3c34b"
}
```

1.2.1.2 共识层请求示例

Request example(With Credential):

```
curl -X GET -H 'Content-Type: application/json' https://your-http-endpoint/your-credential/eth/v1/beacon/genesis
```

Request example(With IAM Token):

```
curl -X GET -H 'Content-Type: application/json' -H 'X-Auth-Token:your-iam-token' https://your-http-endpoint/eth/v1/beacon/genesis
```

Response example:

```
{"data":
{"genesis_time":"1606824023","genesis_validators_root":"0x4b363db94e286120d76eb905340fdd4e54bfe9f06bf33ff6cf5ad27f511bfe95","genesis_fork_version":"0x00000000"}}
```

1.3 应用程序开发介绍

1.3.1 使用 web3.js 发送 JSON-RPC API 请求

Request example:

```
const Web3 = require('web3');
const url = 'https://your-http-endpoint/your-credential'
const web3 = new Web3(new Web3.providers.HttpProvider(url));
web3.eth.getBlockNumber((error, blockNumber) => {
    if(!error) {
        console.log(blockNumber);
    } else {
        console.log(error);
    }
});
```

Response example:

```
{
"jsonrpc": "2.0",
```

```
"id": 1,
"result": "00f3c34b"
}
```

1.3.2 使用 ethers.js 发送 JSON-RPC API 请求

Request example:

```
const ethers = require('ethers');
const url = 'https://your-http-endpoint/your-credential'
const provider = new ethers.providers.JsonRpcProvider(url)
provider.getBlockNumber((error, blockNumber) => {
    if(!error) {
        console.log(blockNumber);
    } else {
        console.log(error);
    }
});
```

Response example:

```
{
"jsonrpc": "2.0",
"id": 1,
"result": "00f3c34b"
}
```

1.4 智能合约开发介绍

1.4.1 什么是智能合约?

智能合约只是一个运行在以太坊链上的一个程序。 它是位于以太坊区块链上一个特定地址的一系列代码（ 函数 ）和数据（ 状态 ）。

智能合约也是一个**以太坊帐户**，我们称之为合约帐户。 这意味着它们有余额，可以成为交易的对象。 但是，他们无法被人操控，他们是被部署在网络上作为程序运行着。 个人用户可以通过提交交易执行智能合约的某一个函数来与智能合约进行交互。 智能合约能像常规合约一样定义规则，并通过代码自动强制执行。 默认情况下，您无法删除智能合约，与它们的交互是不可逆的。

1.5 以太坊 API 列表

为了让软件应用程序与以太坊区块链交互（ 通过读取区块链数据或向网络发送交易 ），它必须连接到以太坊节点。

为此目的，每个**以太坊客户端**都实现了一项 **JSON-RPC 规范**，因此有一套统一的方法可供应用程序依赖，无论具体的节点或客户端实现如何。

JSON-RPC 是一种无状态的、轻量级远程过程调用 (RPC) 协议。 它定义了一些数据结构及其处理规则。 它与传输无关，因为这些概念可以在同一进程，通过接口、超文本传输协议或许多不同的消息传递环境中使用。 它使用 JSON (RFC 4627) 作为数据格式。

相关链接：JSON-RPC 应用程序接口

1.5.1 专享版

1.5.1.1 常用以太坊 API 列表

少数核心 JSON-RPC 方法需要来自以太坊网络的数据，并且整齐地分为三个主要类别：*Gossip*、*State 和 History*。 使用这些部分中的链接跳转到每个方法，或**查看可用以太坊API列表**。

1.5.1.1.1 Gossip 方法

这些方法用于跟踪链头。 这就是交易如何在网络中传播、如何找到进入区块的方式，以及客户端如何发现新区块的方式。

- **eth_blockNumber**
- **eth_sendRawTransaction**

1.5.1.1.2 State 方法

用于报告所有已存储数据的当前状态的方法。 “状态”就像一大块共享内存，包括帐户余额、合约数据和Gas估算。

- **eth_getBalance**
- **eth_getStorageAt**
- **eth_getTransactionCount**
- **eth_getCode**
- **eth_call**
- **eth_estimateGas**

1.5.1.1.3 History 方法

将每个区块的历史记录追溯到创世块。 这就像一个大的附加文件，包括所有区块头、区块体、叔块和交易收据

- **eth_getBlockTransactionCountByHash**
- **eth_getBlockTransactionCountByNumber**
- **eth_getUncleCountByBlockHash**
- **eth_getUncleCountByBlockNumber**
- **eth_getBlockByHash**
- **eth_getBlockByNumber**
- **eth_getTransactionByHash**
- **eth_getTransactionByBlockHashAndIndex**
- **eth_getTransactionByBlockNumberAndIndex**
- **eth_getTransactionReceipt**
- **eth_getUncleByBlockHashAndIndex**
- **eth_getUncleByBlockNumberAndIndex**

1.5.1.2 可用以太坊 API 列表

1.5.1.2.1 执行客户端 API

以太坊官方API介绍：**JSON-RPC 应用程序接口 | ethereum.org**

Go-Ethereum官方API介绍：**JSON-RPC Server | go-ethereum**

API方法	流控值(Full Node / Full Node(Staking Supported))（次/s）		
	4U16G	8U32G	16U64G
debug_traceBlock	10/5	20/10	50/25
debug_traceBlockByHash	10/5	10/5	50/25
debug_traceBlockByNumber	10/5	10/5	15/7
debug_traceCall	1000/500	4000/2000	10000/5000
debug_traceTransaction	50/25	90/45	300/150
eth_blockNumber	7000/3500	30000/15000	60000/30000
eth_call	2000/1000	12000/6000	30000/15000
eth_chainId	3000/1500	20000/10000	50000/25000
eth_createAccessList	200/100	300/150	500/250
eth_estimateGas	700/350	1500/750	5000/2500

API方法	流控值(Full Node / Full Node(Staking Supported))（次/s）		
	4U16G	8U32G	16U64G
eth_feeHistory	根据BLOCKCOUNT * LEN(REWARDPERCENTILES)决定 50：2700/1350 100：2500/1250 200：2300/1150 300：2200/1120 400：1900/950 500：1800/900 600：1500/750 700：1500/750 800：1500/750 900：1000/500 1000：1000/500 2000：700/350 3000：600/300 5000：400/200 10000：200/100	根据BLOCKCOUNT * LEN(REWARDPERCENTILES)决定 50：22000/11000 100：20000/10000 200：10000/5000 300：9500/4750 400：9000/4500 500：8000/4000 600：7000/3500 700：6000/3000 800：5000/2500 900：5000/2500 1000：4000/2000 2000：3000/1500 3000：2000/1000 5000：1000/500 10000：600/300	根据BLOCKCOUNT * LEN(REWARDPERCENTILES)决定 50：42000/21000 100：35000/17500 200：29000/14500 300：24000/12000 400：20000/10000 500：18000/9000 600：15000/7500 700：14000/7000 800：12000/6000 900：11000/5500 1000：10000/5000 2000：5000/2500 3000：3000/1500 5000：2000/1000 10000：1000/500
eth_gasPrice	3000/1500	20000/10000	40000/20000
eth_getBalance	3000/1500	15000/7500	40000/20000

API方法	流控值(Full Node / Full Node(Staking Supported))（次/s）		
	4U16G	8U32G	16U64G
eth_getBlockByHash	• 返回完整的区块对象：200/100 • 不返回完整区块对象：1500/750	• 返回完整的区块对象：600/300 • 不返回完整区块对象：5000/2500	• 返回完整的区块对象：1500/750 • 不返回完整区块对象：16000/8000
eth_getBlockByNumber	• 返回完整的区块对象：300/150 • 不返回完整区块对象：1500/750	• 返回完整的区块对象：600/300 • 不返回完整区块对象：5000/2500	• 返回完整的区块对象：1500/750 • 不返回完整区块对象：20000/10000
eth_getBlockTransactionCountByHash	3000/1500	1000/500	40000/20000
eth_getBlockTransactionCountByNumber	3000/1500	20000/10000	40000/20000
eth_getCode	1000/500	4000/2000	8000/4000
eth_getFilterChanges	400/200	1000/500	2000/1000
eth_getFilterLogs	50/25	1000/500	2000/1000
eth_getLogs	40/20	100/50	200/100
eth_getProof	1000/500	1000/500	3000/1500
eth_getStorageAt	3000/1500	15000/7500	40000/20000
eth_getTransactionByBlockHashAndIndex	3000/1500	15000/7500	40000/20000
eth_getTransactionByBlockNumberAndIndex	2500/1250	15000/7500	40000/20000
eth_getTransactionByHash	600/300	1500/750	4000/2000
eth_getTransactionCount	3000/1500	15000/7500	40000/20000
eth_getTransactionReceipt	500/250	1500/750	3000/1500
eth_getUncleByBlockHashAndIndex	3000/1500	15000/7500	40000/20000
eth_getUncleByBlockNumberAndIndex	3000/1500	15000/7500	40000/20000

API方法	流控值(Full Node / Full Node(Staking Supported))（次/s）		
	4U16G	8U32G	16U64G
eth_getUncleCountByBlockHash	3000/1500	15000/7500	40000/20000
eth_getUncleCountByBlockNumber	3000/1500	15000/7500	40000/20000
eth_getWork	100/50	2000/1000	5500/2750
eth_maxPriorityFeePerGas	3000/1500	15000/7500	40000/20000
eth_newBlockFilter	600/300	800/400	1800/900
eth_newFilter	100/50	500/250	1000/500
eth_newPendingTransactionFilter	20/10	50/25	80/40
eth_sendRawTransaction	500/250	1000/500	2500/1250
eth_subscribe	100/50	1000/500	1000/500
eth_syncing	3000/1500	20000/10000	50000/25000
eth_uninstallFilter	500/250	2000/1000	3000/1500
eth_unsubscribe	100/50	1000/500	1000/500
net_listening	3000/1500	20000/10000	40000/20000
net_version	3000/1500	20000/10000	40000/20000
txpool_inspect	20/10	40/20	90/45
txpool_status	2000/1000	8000/400	15000/7500
web3_clientVersion	3000/1500	20000/10000	40000/20000
web3_sha3	3000/1500	20000/10000	40000/20000

1.5.1.2.2 信标客户端 API

表 1-1 信标客户端 API 列表

API方法	类型	说明	流控值（次/s）		
			4U16G	8U32G	16U64G
/eth/v1/beacon/genesis	GET	获取创世区块信息。	5000/2500	10000/50000	23000/11500
/eth/v1/beacon/states/{state_id}/root	GET	获取索引状态SSZ哈希树根。	3000/1500	6000/3000	13000/6500
/eth/v1/beacon/states/{state_id}/fork	GET	获取索引状态区块链分叉信息。	3000/1500	7000/3500	17000/8500
/eth/v1/beacon/states/{state_id}/finality_checkpoints	GET	获取索引状态finality检查点。	3000/1500	7000/3500	17000/8500
/eth/v1/beacon/states/{state_id}/validators	GET	获取validator的信息。	5/2	5/2	5/2
/eth/v1/beacon/states/{state_id}/validators/{validator_id}	GET	获取validator_id对应的信息。	300/150	600/300	1000/500
/eth/v1/beacon/states/{state_id}/validator_balances	GET	获取validator的余额。	5/2	5/2	5/2
/eth/v1/beacon/states/{state_id}/committees	GET	获取索引状态全部committees。	5/2	6/3	15/7

API方法	类型	说明	流控值（次/s）		
			4U16G	8U32G	16U64G
/eth/v1/beacon/states/{state_id}/sync_committees	GET	获取索引状态全部sync committees。	1100/550	2800/1400	5000/2500
/eth/v1/beacon/headers	GET	获取区块头信息。	1800/900	4000/2000	8000/4000
/eth/v1/beacon/headers/{block_id}	GET	通过区块id获取区块头信息。	1400/700	2000/1000	6000/3000
/eth/v2/beacon/blocks/{block_id}	GET	通过区块id获取区块信息。	50/25	90/45	300/150
/eth/v1/beacon/blocks/{block_id}/root	GET	通过区块id获取区块根信息。	5000/2500	9000/4500	22000/11000
/eth/v1/beacon/blocks/{block_id}/attestations	GET	通过区块id获取区块验证信息。	300/150	700/350	1800/900
/eth/v1/beacon/rewards/blocks/{block_id}	GET	获取区块奖励信息。	90/45	110/55	120/60
/eth/v1/beacon/rewards/attestations/{epoch}	POST	获取对应epoch的validator的证明奖励。	5/2	5/2	5/2
/eth/v1/beacon/blinded_blocks/{block_id}	GET	通过区块id获取盲块。	300/150	600/300	1400/700

API方法	类型	说明	流控值（次/s）		
			4U16G	8U32G	16U64G
/eth/v1/beacon/pool/attestations	GET	获取操作池中的attestations。	2200/1100	3000/1500	4000/2000
/eth/v1/beacon/pool/attester_slashings	GET	获取操作池中的AttesterSlashings。	5000/2500	10000/50000	23000/11500
/eth/v1/beacon/pool/proposer_slashings	GET	获取操作池中的ProposerSlashings。	6000/3000	11000/5500	24000/12000
/eth/v1/beacon/pool/voluntary_exits	GET	获取操作池中的SignedVuntaryExit。	5000/2500	11000/5500	24000/12000
/eth/v1/beacon/pool/bls_to_execution_changes	GET	获取从BLS到节点已知但不一定合并到任何区块中的执行层更改。	6000/3000	11000/5500	24000/12000
/eth/v1/builder/states/{state_id}/expected_withdrawals	GET	获取在指定状态上构建的区块要包括的提款。	2000/1000	3000/1500	6000/3000
/eth/v1/config/fork_schedule	GET	获取所有分叉信息。	6000/3000	10000/50000	21000/10500
/eth/v1/config/spec	GET	获取节点相关联的配置信息。	1700/850	4000/2000	9000/4500

API方法	类型	说明	流控值（次/s）		
			4U16G	8U32G	16U64G
/eth/v1/config/deposit_contract	GET	获取节点相关联的Eth1存款合约地址以及链id。	7000/3500	10000/50000	24000/120000
/eth/v2/debug/beacon/states/{state_id}	GET	获取完整的BeaconState对象。	5/2	5/2	5/2
/eth/v2/debug/beacon/heads(Deprecated)	GET	获取所有链头信息。	6000/3000	10000/50000	13000/6500
/eth/v1/debug/fork_choice	GET	获取fork choice数组。	600/300	1000/500	2000/1000
/eth/v1/events	GET	订阅beacon node事件。	-	-	-
/eth/v1/node/version	GET	获取beacon node版本。	5000/2500	10000/50000	23000/115000
/eth/v1/node/syncing	GET	获取beacon node同步情况。	5000/2500	10000/50000	23000/115000
/eth/v1/node/health	GET	获取beacon node健康检查结果。	5000/2500	11000/5500	24000/120000
/eth/v1/validator/duties/attester/{epoch}	POST	获取validator的duties。	5/2	5/2	5/2

API方法	类型	说明	流控值（次/s）		
			4U16G	8U32G	16U64G
/eth/v1/validator/duties/proposer/{epoch}	GET	获取区块提案者的duties。	5/2	5/2	5/2
/eth/v1/validator/duties/sync/{epoch}	POST	获取同步委员会的的duties。	5/2	5/2	5/2
/eth/v1/validator/aggregate_attestation	GET	获取聚合证明。	4000/2000	8000/4000	15000/7500
/eth/v1/validator/sync_committee_contribution	GET	生成同步委员会贡献。	5000/2500	11000/5500	18000/9000
/eth/v1/validator/liveness/{epoch}	POST	返回是否在网络上观察到validator。	5/2	5/2	5/2

1.5.1.3 可用 Validator 所需 API 列表

表 1-2 可用验证器所需 API 列表

API方法	类型	说明
/eth/v1/node/version	GET	获取beacon node版本。
/eth/v1/beacon/genesis	GET	获取创世区块信息。
/eth/v1/beacon/states/{state_id}/fork	GET	获取索引状态区块链分叉信息。
/eth/v1/beacon/states/{state_id}/validators/{validator_id}	GET	获取validator_id对应的信息。
/eth/v1/beacon/blinded_blocks	POST	发布一个签名的区块。
/eth/v2/beacon/blinded_blocks	POST	发布一个签名的区块。
/eth/v1/beacon/blocks	POST	发布一个签名的区块。

API方法	类型	说明
/eth/v2/beacon/blocks	POST	发布一个签名的区块。
/eth/v1/beacon/blocks/{block_id}/root	GET	通过区块id获取区块根信息。
/eth/v1/beacon/pool/attestations	POST	提交证明对象到节点。
/eth/v1/beacon/pool/sync_committees	POST	提交同步委员会签名到节点。
/eth/v1/node/syncing	GET	获取beacon node同步情况。
/eth/v1/config/spec	GET	获取节点相关联的配置信息。
/eth/v1/validator/duties/attester/{epoch}	POST	获取validator的duties。
/eth/v1/validator/duties/proposer/{epoch}	GET	获取区块提案者的duties。
/eth/v1/validator/duties/sync/{epoch}	POST	获取同步委员会的的duties。
/eth/v2/validator/blocks/{slot} (Deprecated)	POST	生成无签名区块。
/eth/v3/validator/blocks/{slot}	GET	生成无签名区块。
/eth/v1/validator/blinded_blocks/{slot} (Deprecated)	GET	生成一个无签名盲块。
/eth/v1/validator/attestation_data	GET	生成证明数据。
/eth/v1/validator/aggregate_attestation	GET	获取聚合证明。
/eth/v1/validator/aggregate_and_proofs	POST	发布多个聚合证明。
/eth/v1/validator/beacon_committee_subscriptions	POST	通知beacon node准备委员会subnet。
/eth/v1/validator/sync_committee_subscriptions	POST	订阅同步委员会subnet。
/eth/v1/validator/sync_committee_contribution	GET	生成同步委员会贡献。
/eth/v1/validator/contribution_and_proofs	POST	发布多个贡献和证明。

API方法	类型	说明
/eth/v1/validator/prepare_beacon_proposer	POST	为提案者准备信标节点。
/eth/v1/events	GET	订阅beacon node事件。
/eth/v1/validator/register_validator	POST	注册validator
/eth/v1/config/deposit_contract	GET	获取存款合约地址
/eth/v1/config/fork_schedule	GET	获取所有分叉信息。
/eth/v1/beacon/states/{state_id}/validators	GET	获取validator的信息。

1.5.2 共享版

共享版支持的JSON-RPC API请求列表。

1.5.2.1 eth_blocknumber

简介

返回区块链的最新区块号。该API所消耗的计算单元为15。

参数说明

此方法不接受任何参数。

返回值

十六进制编码的最新区块号。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"eth_blockNumber","params":[],"id":1,"jsonrpc":"2.0"}'
```

1.5.2.2 eth_getBlockByNumber

简介

返回与给定的区块号匹配的区块信息。该API所消耗的计算单元为49。

参数说明

参数	类型	说明
区块编号	String	十六进制的区块编号，或者是字符串 "earliest"、"latest"、"pending"。
交易详细信息标志	Bool	当此值为 true 时，该方法返回完整的交易对象，否则，它仅返回交易的哈希。

返回值

- Object - 区块对象，如果未找到区块，则为 null。区块对象包含以下字段：
  - number: 编码为十六进制的请求块的块号。如果是一个处于待处理（ Pending ）状态的区块，则为空。
  - hash: 区块的哈希。如果是一个处于待处理（ Pending ）状态的区块，则为空。
  - parentHash: 父区块的哈希。
  - nonce: 生成的工作量证明的哈希值。如果是一个处于待处理（ Pending ）状态的区块，则为空。
  - sha3Uncles: 区块中叔区块数据的 SHA3。
  - logsBloom: 区块日志的布隆过滤器。如果是一个处于待处理（ Pending ）状态的区块，则为空。
  - transactionsRoot: 区块中交易树的根。
  - stateRoot： 区块的最终状态树的根。
  - receiptsRoot： 区块的收据树的根。
  - miner： 获得采矿奖励的受益人的地址。
  - difficulty：此区块的难度。
  - totalDifficulty：直到这个区块时，链的总难度。
  - extraData：此区块的“额外数据”字段。
  - size：此区块的大小（以字节为单位）。
  - gasLimit：此区块中允许的最大gas。
  - gasUsed：此区块中所有交易的总使用gas。
  - timestamp：整理区块时的 unix 时间戳。
  - transactions：交易对象的数组，或 32 字节的交易哈希，具体取决于最后一个给定的参数。
  - uncles：叔区块哈希数组。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"eth_getBlockByNumber","params":["0xc5043f",false],"id":1,"jsonrpc":"2.0"}'
```

1.5.2.3 eth_getUncleByBlockNumberAndIndex

简介

按区块编号和叔区块索引位置返回有关叔区块的信息。该API所消耗的计算单元为17。

参数说明

参数	类型	说明
区块编号或标签	String	想要查询的区块的十六进制的编号，或者是字符串 "earliest"、"latest"、"pending"。
叔区块的索引位置	String	叔区块的十六进制的区块编号。

返回值

- Object - 区块对象，如果未找到区块，则为 null。区块对象包含以下字段：
  - number: 编码为十六进制的请求块的块号。如果是一个处于待处理（ Pending ）状态的区块，则为空。
  - hash: 区块的哈希。如果是一个处于待处理（ Pending ）状态的区块，则为空。
  - parentHash: 父区块的哈希。
  - nonce: 生成的工作量证明的哈希值。如果是一个处于待处理（ Pending ）状态的区块，则为空。
  - sha3Uncles: 区块中叔区块数据的 SHA3。
  - logsBloom: 区块日志的布隆过滤器。如果是一个处于待处理（ Pending ）状态的区块，则为空。
  - transactionsRoot: 区块中交易树的根。
  - stateRoot： 区块的最终状态树的根。
  - receiptsRoot： 区块的收据树的根。
  - miner： 获得采矿奖励的受益人的地址。
  - difficulty：此区块的难度。
  - totalDifficulty： 直到这个区块时，链的总难度。
  - extraData：此区块的“额外数据”字段。
  - size：此区块的大小（以字节为单位）。
  - gasLimit：此区块中允许的最大gas。
  - gasUsed：此区块中所有交易的总使用gas。
  - timestamp：整理区块时的 unix 时间戳。
  - transactions：交易对象的数组，或 32 字节的交易哈希，具体取决于最后一个给定的参数。
  - uncles：叔区块哈希数组。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"eth_getUncleByBlockNumberAndIndex","params":["latest","0x0"],"id":1,"jsonrpc":"2.0"}'
```

1.5.2.4 eth_getUncleByBlockHashAndIndex

简介

按区块哈希和叔区块索引位置返回有关叔区块的信息。该API所消耗的计算单元为17。

参数说明

参数	类型	说明
区块哈希	String	想要查询的区块的哈希值。
叔区块的索引位置	String	叔区块的十六进制的区块编号。

返回值

- Object - 区块对象，如果未找到区块，则为 null。区块对象包含以下字段：
  - number: 编码为十六进制的请求块的块号。如果是一个处于待处理（ Pending ）状态的区块，则为空。
  - hash: 区块的哈希。如果是一个处于待处理（ Pending ）状态的区块，则为空。
  - parentHash: 父区块的哈希。
  - nonce: 生成的工作量证明的哈希值。如果是一个处于待处理（ Pending ）状态的区块，则为空。
  - sha3Uncles: 区块中叔区块数据的 SHA3。
  - logsBloom: 区块日志的布隆过滤器。如果是一个处于待处理（ Pending ）状态的区块，则为空。
  - transactionsRoot: 区块中交易树的根。
  - stateRoot： 区块的最终状态树的根。
  - receiptsRoot： 区块的收据树的根。
  - miner： 获得采矿奖励的受益人的地址。
  - difficulty：此区块的难度。
  - totalDifficulty： 直到这个区块时，链的总难度。
  - extraData：此区块的“额外数据”字段。
  - size：此区块的大小（以字节为单位）。
  - gasLimit：此区块中允许的最大gas。
  - gasUsed：此区块中所有交易的总使用gas。
  - timestamp：整理区块时的 unix 时间戳。

- transactions：交易对象的数组，或 32 字节的交易哈希，具体取决于最后一个给定的参数。
- uncles：叔区块哈希数组。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"eth_getUncleByBlockHashAndIndex","params":
["0xc6ef2fc5426d6ad6fd9e2a26abeab0aa2411b7ab17f30a99d3cb96aed1d1055b",
"0x0"],"id":1,"jsonrpc":"2.0"}'
```

1.5.2.5 eth_getUncleCountByBlockNumber

简介

返回与给定区块编号匹配的区块中叔区块的数量。该API所消耗的计算单元为17。

参数说明

参数	类型	说明
区块编号	String	想要查询的区块的十六进制的编号。

返回值

区块中叔区块的数量，以十六进制为编码。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"eth_getUncleCountByBlockNumber","params":["0xc5043f"],"id":1,"jsonrpc":"2.0"}'
```

1.5.2.6 eth_getUncleCountByBlockHash

简介

返回与给定区块哈希匹配的区块中叔区块的数量。该API所消耗的计算单元为17。

参数说明

参数	类型	说明
区块哈希	String	想要查询的区块的哈希值。

返回值

区块中叔区块的数量，以十六进制为编码。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"eth_getUncleCountByBlockHash","params":
["0x81e807e7a6031d9f103eeee2a2edc5994c3432ee1e3227c66ff78eef30ea1dec"],"id":1,"jsonrpc":"2.0"}'
```

1.5.2.7 eth_getBlockByHash

简介

返回与给定区块哈希匹配的区块的信息。该API所消耗的计算单元为47。

参数说明

参数	类型	说明
区块哈希	String	想要查询的区块的哈希值。
事务详细信息标志	Bool	当此值为 true 时，该方法返回完整的交易对象，否则，它仅返回交易的哈希。

返回值

- Object - 区块对象，如果未找到区块，则为 null。区块对象包含以下字段：
  - number: 编码为十六进制的请求块的块号。如果是一个处于待处理（ Pending ）状态的区块，则为空。
  - hash: 区块的哈希。如果是一个处于待处理（ Pending ）状态的区块，则为空。
  - parentHash: 父区块的哈希。
  - nonce: 生成的工作量证明的哈希值。如果是一个处于待处理（ Pending ）状态的区块，则为空。
  - sha3Uncles: 区块中叔区块数据的 SHA3。
  - logsBloom: 区块日志的布隆过滤器。如果是一个处于待处理（ Pending ）状态的区块，则为空。
  - transactionsRoot: 区块中交易树的根。
  - stateRoot： 区块的最终状态树的根。
  - receiptsRoot： 区块的收据树的根。
  - miner： 获得采矿奖励的受益人的地址。
  - difficulty：此区块的难度。
  - totalDifficulty： 直到这个区块时，链的总难度。
  - extraData：此区块的“额外数据”字段。
  - size：此区块的大小（以字节为单位）。
  - gasLimit：此区块中允许的最大gas。
  - gasUsed：此区块中所有交易的总使用gas。

- timestamp：整理区块时的 unix 时间戳。
- transactions：交易对象的数组，或 32 字节的交易哈希，具体取决于最后一个给定的参数。
- uncles：叔区块哈希数组。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"eth_getBlockByHash","params":
["0x81e807e7a6031d9f103eeee2a2edc5994c3432ee1e3227c66ff78eef30ea1dec",false],"id":1,"jsonrpc":"2.0"}'
```

1.5.2.8 eth_getTransactionByHash

简介

根据交易哈希返回有关交易的信息。该API所消耗的计算单元为25。

参数说明

参数	类型	说明
交易哈希	String	要查询的交易的哈希值。

返回值

- Object - 交易对象，如果未找到交易，则为 null。交易对象包含以下字段：
  - blockHash：此交易所在的区块的哈希值。当它是待处理的（Pending）日志时为 null
  - blockNumber：此交易所在的区块号。当它是待处理的（Pending）日志时为 null
  - from：发件人的地址
  - gas：发送方提供的gas，编码为十六进制
  - gasPrice：发件人提供的 wei 格式的gas价格，编码为十六进制
  - maxFeePerGas：交易中设置的每种gas的最高值
  - maxPriorityFeePerGas：交易中设置的最高优先级gas
  - hash：交易的哈希值
  - input：与交易一起发送的数据
  - nonce：发送方在此交易之前进行的交易数，编码为十六进制
  - to：接收方的地址。当它是合约创建交易时为 null
  - transactionIndex：从中创建日志的交易索引位置的整数。当它是待处理的（Pending）日志时为 null
  - value：以 wei 为单位的十六进制编码的转账数额
  - type：交易类型
  - accessList：交易计划访问的地址和存储密钥的列表

- chainId：交易的链 ID（如果有）
- v：签名的标准化 V 字段
- r：签名的 R 字段
- s：签名的 S 字段

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"eth_getTransactionByHash","params":
["0xb142342a7fd70602b7a0ba3688a41bfcbb4fbc3490c252ca48af2594619d220c"],"id":1,"jsonrpc":"2.0"}'
```

1.5.2.9 eth_getTransactionCount

简介

返回从某一地址发送的交易数。该API所消耗的计算单元为26。

参数说明

参数	类型	说明
地址	String	要检查的交易计数的地址。
区块编号	String	想要查询的区块的十六进制的编号，或者是字符串"earliest"、"latest"、"pending"。

返回值

从地址发送的十六进制编码的交易数量

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"eth_getTransactionCount","params":
["0x8D97689C9818892B700e27F316cc3E41e17fBeb9", "latest"],"id":1,"jsonrpc":"2.0"}'
```

1.5.2.10 eth_getTransactionByBlockHashAndIndex

简介

返回给定交易哈希和交易索引位置的交易信息。该API所消耗的计算单元为20。

参数说明

参数	类型	说明
交易哈希	String	想要查询的交易的哈希值。
索引	String	编码为十六进制的交易索引位置。

返回值

- Object - 交易对象，如果未找到交易，则为 null。交易对象包含以下字段：
  - blockHash：此交易所在的区块的哈希值。当它是待处理的（Pending）日志时为 null
  - blockNumber：此交易所在的区块号。当它是待处理的（Pending）日志时为 null
  - from：发件人的地址
  - gas：发送方提供的gas，编码为十六进制
  - gasPrice：发件人提供的以 wei 为单位的gas价格，编码为十六进制
  - maxFeePerGas：交易中设置的每种gas的最高值
  - maxPriorityFeePerGas：交易中设置的最高优先级gas
  - hash：交易的哈希值
  - input：与交易一起发送的数据
  - nonce：发送方在此交易之前进行的交易数，编码为十六进制
  - to：接收方的地址。当它是合约创建交易时为 null
  - transactionIndex：从中创建日志的交易索引位置的整数。当它是待处理的（Pending）日志时为 null
  - value：以 wei 为单位的十六进制编码的转账数额
  - type：交易类型
  - accessList：交易计划访问的地址和存储密钥的列表
  - chainId：交易的链 ID（如果有）
  - v：签名的标准化 V 字段
  - r：签名的 R 字段
  - s：签名的 S 字段

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"eth_getTransactionByBlockHashAndIndex","params":
["0x81e807e7a6031d9f103eeee2a2edc5994c3432ee1e3227c66ff78eef30ea1dec","0x0"],"id":1,"jsonrpc":"2.0"}
'
```

1.5.2.11 eth_getTransactionByBlockNumberAndIndex

简介

返回给定区块号和交易索引位置的交易信息。该API所消耗的计算单元为20。

参数说明

参数	类型	说明
区块编号	String	想要查询的区块的十六进制的编号，或者是字符串"earliest"、"latest"、"pending"。
索引	String	编码为十六进制的交易索引位置。

返回值

- Object - 交易对象，如果未找到交易，则为 null。交易对象包含以下字段：
  - blockHash：此交易所在的区块的哈希值。当它是待处理的（ Pending ）日志时为 null
  - blockNumber：此交易所在的区块号。当它是待处理的（ Pending ）日志时为 null
  - from：发件人的地址
  - gas：发送方提供的gas，编码为十六进制
  - gasPrice：发件人提供的以 wei 为单位的gas价格，编码为十六进制
  - maxFeePerGas：交易中设置的每种gas的最高值
  - maxPriorityFeePerGas：交易中设置的最高优先级gas
  - hash：交易的哈希值
  - input：与交易一起发送的数据
  - nonce：发送方在此交易之前进行的交易数，编码为十六进制
  - to：接收方的地址。当它是合约创建交易时为 null
  - transactionIndex：从中创建日志的交易索引位置的整数。当它是待处理的（ Pending ）日志时为 null
  - value：以 wei 为单位的十六进制编码的转账数额
  - type：交易类型
  - accessList：交易计划访问的地址和存储密钥的列表
  - chainId：交易的链 ID（如果有）
  - v：签名的标准化 V 字段
  - r：签名的 R 字段
  - s：签名的 S 字段

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"eth_getTransactionByBlockNumberAndIndex","params":["0xc5043f",
"0x0"],"id":1,"jsonrpc":"2.0"}'
```

1.5.2.12 eth_getBlockTransactionCountByHash

简介

返回与给定块哈希匹配的区块的交易数。该API所消耗的计算单元为20。

参数说明

参数	类型	说明
区块哈希	String	想要查询的区块的哈希值。

返回值

以十六进制格式表示的所查询区块中的交易数。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"eth_getBlockTransactionCountByHash","params":
["0x81e807e7a6031d9f103eeee2a2edc5994c3432ee1e3227c66ff78eef30ea1dec"],"id":1,"jsonrpc":"2.0"}'
```

1.5.2.13 eth_getBlockTransactionCountByNumber

简介

返回与给定区块编号匹配的区块的交易数。该API所消耗的计算单元为20。

参数说明

参数	类型	说明
区块编号	String	想要查询的区块的十六进制的编号，或者是字符串"earliest"、"latest"、"pending"。

返回值

以十六进制格式表示的所查询区块中的交易数。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"eth_getBlockTransactionCountByNumber","params":["0xc5043f"],"id":1,"jsonrpc":"2.0"}'
```

1.5.2.14 eth_getTransactionReceipt

简介

通过交易哈希返回交易的收据。该API所消耗的计算单元为15。

参数说明

参数	类型	说明
交易哈希	String	想要查询的交易的哈希值。

返回值

- Object - 交易收据对象，如果未找到交易收据，则为 null。交易收据对象包含以下字段：
  - blockHash：此交易所在的区块的哈希值
  - blockNumber：添加此交易的区块号，编码为十六进制
  - contractAddress：为创建合约创建的合约地址，如果并非合约创建则为空
  - cumulativeGasUsed：在区块中执行此交易时使用的总gas
  - effectiveGasPrice：为每单位gas支付的总基本费用加上额外交易费
  - from：源地址
  - gasUsed：仅此特定交易使用的gas
  - logs：生成此交易的日志对象数组
    - address：生成此日志的地址
    - topics：索引日志参数的 0 到 4 个 32 字节 DATA 的数组。在 Solidity 中，第一个主题是事件签名的哈希值（例如 Deposit(address, bytes32, uint256)），除非您使用匿名说明符声明事件
    - data：日志的 32 字节非索引参数
    - blockNumber：此日志所在的块号
    - transactionHash：从中创建此日志的交易的哈希。如果日志处于待处理（ Pending ）状态，则为 null
    - transactionIndex：从中创建此日志的交易索引位置。如果日志处于待处理（ Pending ）状态，则为 null
    - blockHash：此日志所在的块的哈希值
    - logIndex：编码为十六进制的块中对数索引位置的整数。如果日志处于待处理（ Pending ）状态，则为 null

    - removed：如果日志由于链重组而被删除，则为 true，如果它是有效的日志，则为 false。
  - logsBloom：用于检索相关日志的布隆过滤器
  - status：1（成功）或 0（失败），编码为十六进制
  - to：接收方的地址。当它是合约创建交易时为 null
  - transactionHash：交易的哈希值
  - transactionIndex：编码为十六进制的块中的交易索引位置
  - type：值的类型

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"eth_getTransactionReceipt","params":
["0x6d755989f51032147484162c4dc3d6550552dbd8d3b094fe3c221bfa3c5942b2"],"id":1,"jsonrpc":"2.0"}'
```

1.5.2.15 eth_sendRawTransaction

简介

创建新的消息调用交易或为签名交易创建合约。该API所消耗的计算单元为250。

参数说明

参数	类型	说明
签名后的交易数据	String	使用您的私钥签名的交易。

返回值

交易哈希值，如果交易尚不可用，则为零哈希值。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"jsonrpc":"2.0","method":"eth_sendRawTransaction","params":["signed transaction"],"id":1}'
```

1.5.2.16 eth_call

简介

立即执行新的消息调用，而不在区块链上创建交易。该API所消耗的计算单元为30。

参数说明

包含交易的相关字段以及区块编号两部分。

参数	类型	说明
from	String	可选参数，发送交易的地址。
to	String	交易发送到的地址。
gas	Integer	可选参数，为交易执行提供的gas的整数。
gasPrice	Integer	可选参数，用于每个付费gas的gasPrice整数，编码为十六进制。
value	Integer	可选参数，与此交易一起发送的代币的数值，编码为十六进制。
data	String	可选参数，方法签名和编码参数的哈希值。有关更多信息，请参阅 Solidity 文档中的合约 ABI 描述。
区块编号	String	想要查询的区块的十六进制的编号，或者是字符串"earliest"、"latest"、"pending"。

返回值

执行合约方法的返回值。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"eth_call","params":
[{"from":null,"to":"0x6b175474e89094c44da98b954eedeac495271d0f","data":"0x70a08231000000000000000000000000006E0d01A76C3Cf4288372a29124A26D4353EE51BE"}, "latest"],"id":1,"jsonrpc":"2.0"}'
```

1.5.2.17 eth_createAccessList

简介

基于给定的交易对象创建 EIP2930 类型 accessList。返回交易读取和写入的地址以及存储密钥列表，但是发送者帐户和预编译除外。该API所消耗的计算单元为300。

参数说明

包含交易的相关字段以及区块编号两部分。

参数	类型	说明
from	String	发送交易的地址
to	String	交易发送到的地址
gas	Integer	为交易执行提供的gas的整数
gasPrice	Integer	用于每个付费Gas的gasPrice整数，编码为十六进制
value	Integer	与此交易一起发送的代币的数值，编码为十六进制
data	String	方法签名和编码参数的哈希值。有关更多信息，请参阅 Solidity 文档中的合约 ABI 描述
区块编号	String	想要查询的区块的十六进制的编号，或者是字符串"earliest"、"latest"、"pending"。

返回值

返回除发送者帐户和预编译之外的所有交易读取和写入的地址以及存储密钥列表，以及添加访问列表时消耗的估计gas。

- accessList: 具有以下字段的对象列表:
  - address: 交易要访问的地址。
  - storageKeys: 交易要访问的存储密钥。
- gasUsed: 十六进制字符串，表示交易的大致gas成本（ 如果包含访问列表 ）。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
    -X POST \
    -H "Content-Type: application/json" \
    -d '{"method":"eth_createAccessList","params":[{"from":
"0xaeA8F8f781326bfE6A7683C2BD48Dd6AA4d3Ba63", "data": "0x608060806080608155"},
"pending"],"id":1,"jsonrpc":"2.0"}'
```

1.5.2.18 eth_estimateGas

简介

返回给定交易的所消耗的Gas的估计值。该API所消耗的计算单元为90。

参数说明

与 eth_call 的参数一致，但所有属性都是可选的。 如果没有指定Gas限制，geth 将使用来自待处理区块的区块Gas限制作为上限。 因此，当所需Gas数量高于待处理区块的Gas限制时，返回的估算值可能不足以执行调用/交易。

参数	类型	说明
from	String	发送交易的地址。
to	String	交易发送到的地址。
Gas	Integer	为交易执行提供的Gas的整数。
gasPrice	Integer	用于每个付费gas的gasPrice整数，编码为十六进制。
value	Integer	与此交易一起发送的代币的数值，编码为十六进制。
data	String	方法签名和编码参数的哈希值。有关更多信息，请参阅 Solidity 文档中的合约 ABI 描述。
区块编号	String	想要查询的区块的十六进制的编号，或者是字符串"earliest"、"latest"、"pending"。

返回值

交易所消耗Gas的预计数量。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"eth_estimateGas","params":
[{"from":"0x8D97689C9818892B700e27F316cc3E41e17fBeb9","to":"0xd3CdA913deB6f67967B99D67aCDFa1712C293601","value":"0x186a0"}],"id":1,"jsonrpc":"2.0"}'
```

1.5.2.19 eth_feeHistory

简介

返回历史消耗的Gas信息的集合。该API所消耗的计算单元为16。

参数说明

参数	类型	说明
区块数量	String/Integer	请求范围内的块数。在单个查询中可以请求 1 到 1024 个块。如果不是所有块都可用，它将返回小于请求的范围。
最新区块编号	String	想要查询的区块的十六进制的编号，或者是字符串 "earliest"、"latest"、"pending"。
奖励百分位数	Integer	可选参数，单调递增的百分位值列表，从每个区块的每种 gas 的有效优先费中采样，按升序排列，并按所使用的 gas 进行加权。

返回值

- oldestBlock：以十六进制数表示的返回范围中最早的区块编号。
- baseFeePerGas：数组，内容为每个 gas 的一系列区块基本费用，包括额外的区块值。额外的值是返回范围中最新块之后的下一个块。对于EIP-1559之前创建的块返回零。
- gasUsedRatio：数组，内容为每个区块gas使用比率。计算方式为gasUsed和gasLimit的比率。
- reward：来自单个区块的每个Gas数据点的有效优先费数组。如果块为空，则返回所有零。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  -d '{"id": 1, "jsonrpc": "2.0", "method": "eth_feeHistory", "params": ["0x5", "latest", [20,30]] }'
```

1.5.2.20 eth_maxPriorityFeePerGas

简介

返回每个Gas的费用，这是您可以支付多少优先费用或“小费”的估计，以获得当前区块中包含的交易。该API所消耗的计算单元为16。

参数说明

此方法不接受任何参数。

返回值

十六进制代码编码的当前区块中包含交易的每项 gas 费用。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
    -X POST \
    -H "Content-Type: application/json" \
    -d '{"jsonrpc":"2.0","method":"eth_maxPriorityFeePerGas","id":1}'
```

1.5.2.21 eth_gasPrice

简介

返回当前的Gas价格（以 wei 为单位）。该API所消耗的计算单元为19。

参数说明

此方法不接受任何参数。

返回值

以十六进制表示的当前Gas的价格，以 wei 为单位。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  -d '{"jsonrpc":"2.0","method":"eth_gasPrice","params": [],"id":1}'
```

1.5.2.22 eth_getBalance

简介

返回给定地址的帐户余额。该API所消耗的计算单元为23。

参数说明

参数	类型	说明
地址	String	表示用于检查余额的地址
区块编号	String	想要查询的区块的十六进制的编号，或者是字符串"earliest"、"latest"、"pending"。

返回值

以十六进制编码的给定地址帐户中当前余额，以wei为单位。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  -d '{"jsonrpc":"2.0","method":"eth_getBalance","params":
["0xc94770007dda54cF92009BFF0dE90c06F603a09f", "latest"],"id":1}'
```

1.5.2.23 eth_subscribe

简介

为特定事件创建新订阅。节点返回订阅 ID。对于与订阅匹配的每个事件，将发送包含相关数据的通知以及订阅 ID。该API所消耗的计算单元为10。

参数说明

参数	类型	说明
事件类型	String	指定要侦听的事件类型。
可选参数	String	要包含的可选参数，用于描述要侦听的事件类型，包括newHeads、newPendingTransactions、logs。

返回值

当订阅处于活动状态时，您将收到格式化为如下对象的事件：

事件对象：

- jsonrpc：始终为“2.0”。
- method：始终“eth_subscription”。
- params：具有以下字段的对象：
  - subscription：创建此订阅的调用返回的订阅 ID。此 ID 将附加到所有收到的事件，也可用于使用eth_unsubscribe。
  - result：内容因事件类型而异的对象。

请求样式

```
wscat -c wss://your-http-endpoint/v1/<API-KEY> -x '{"jsonrpc":"2.0", "id": 1, "method": "eth_subscribe",
"params": ["logs"]}'
```

1.5.2.24 eth_unsubscribe

简介

通过使用订阅 ID 调用此方法来取消订阅。它返回一个布尔值，指示订阅已成功取消。该API所消耗的计算单元为10。

参数说明

参数	类型	说明
订阅ID	String	要取消订阅的订阅的 ID。

返回值

如果订阅已成功取消，返回True，否则返回False。

请求样式

```
wscat -c wss://your-http-endpoint/v1/<API-KEY> -x '{"jsonrpc":"2.0", "id": 1, "method": "eth_unsubscribe",
"params": ["0x9cef478923ff08bf67fde6c64013158d"]}'
```

1.5.2.25 eth_getStorageAt

简介

返回给定地址的存储位置的值。该API所消耗的计算单元为23。

参数说明

参数	类型	说明
地址	String	表示存储地址（ 20 字节 ）的字符串。
存储位置	String	存储位置的十六进制代码。
区块编号	String	想要查询的区块的十六进制的编号，或者是字符串 "earliest"、"latest"、"pending"。

返回值

所提供存储位置的值。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  -d '{"jsonrpc":"2.0","method":"eth_getStorageAt","params":
["0x295a70b2de5e3953354a6a8344e616ed314d7251",
"0x6661e9d6d8b923d5bbaab1b96e1dd51ff6ea2a93520fdc9eb75d059238b8c5e9", "0x65a8db"],"id":1}'
```

1.5.2.26 eth_getCode

简介

返回给定地址处智能合约的已编译字节代码（如果有）。该API所消耗的计算单元为22。

参数说明

参数	类型	说明
地址	String	表示存储地址（20 字节）的字符串，从中获取编译后的字节码。
区块编码	String	想要查询的区块的十六进制的编号，或者是字符串"earliest"、"latest"、"pending"。

返回值

给定地址处智能合约的已编译字节代码。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  -d '{"jsonrpc":"2.0","method":"eth_getCode","params":
["0x06012c8cf97bead5deae237070f9587f8e7a266d", "0x65a8db"],"id":1}'
```

1.5.2.27 eth_getProof

简介

返回指定账户的账户和存储值，包括 Merkle 证明。该API所消耗的计算单元为43。

参数说明

参数	类型	说明
账户地址	String	表示存储地址（20 字节）的字符串，从中获取编译后的字节码。
存储键	Array	要验证和包含的 32 字节存储键值的数组。

参数	类型	说明
区块编号	String	想要查询的区块的十六进制的编号，或者是字符串"earliest"、"latest"、"pending"。

返回值

- address：与账户相关的地址。
- accountProof：RLP 序列化 MerkleTree-Nodes 的数组，从 stateRoot-Node 开始，遵循 SHA3（地址）的路径作为键。
- balance：当前余额的十六进制，以 wei 为单位。
- codeHash：账户代码的 32 字节哈希值。
- nonce：账户的随机数。
- storageHash: 32 字节。StorageRoot 的 SHA3。所有存储都将从这里开始提供 Merkle 证明rootHash。
- storageProof：请求的存储条目数组。每个条目都是一个具有以下属性的对象：
  - key：请求的存储密钥。
  - value：存储值。
  - proof：RLP 序列化 MerkleTree-Nodes 的数组，从 storageHash-Node 开始，遵循 SHA3（密钥）的路径作为路径。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  -d '{"jsonrpc": "2.0","method": "eth_getProof","id": 1,"params":
["0x7F0d15C7FAae65896648C8273B6d7E43f58Fa842",
["0x56e81f171bcc55a6ff8345e692c0f86e5b48e01b996cadc001622fb5e363b421"], "latest"]}'
```

1.5.2.28 eth_getLogs

简介

返回与给定过滤器对象匹配的所有日志的数组。该API所消耗的计算单元为75。

参数说明

参数	类型	说明
address	String	可选参数，合约地址（20字节）或日志应源自的地址列表。

参数	类型	说明
fromBlock	String	可选参数，默认为“latest”，十六进制区块号，或字符串latest,earliest或pending。
toBlock	String	可选参数，默认为“latest”，十六进制区块号，或字符串latest,earliest或pending。
topics	String	可选参数，32 字节 DATA 主题数组。主题与顺序相关。
blockhash	String	可选参数，将返回的日志限制为 32 字节哈希中引用的单个块blockHash。使用blockHash相当于设置fromBlock和 toBlock中引用的区块均为blockHash所对应的区块。如果blockHash出现在过滤条件中，则fromBlock和toBlock都不能够被设置。

返回值

日志对象数组，如果自上次轮询以来没有任何更改，则为空数组。日志对象包含以下键及其值：

- removed：若日志由于链重组而被删除，则返回true。如果它是有效的日志，则返回false。
- logIndex：块中日志索引位置的十六进制。当它是待处理（Pending）日志时返回NULL。
- transactionIndex：创建日志的事务索引位置的十六进制。当它是待处理（Pending）日志时返回NULL。
- transactionHash: 32 字节。创建此日志的事务的哈希值。当它是待处理（Pending）日志时返回NULL。
- blockHash: 32 字节。该日志所在块的哈希值，当它是待处理（Pending）日志时返回NULL。
- blockNumber：该日志所在的块号，当它是待处理（Pending）日志时返回NULL。
- address: 20 字节。该日志的来源地址。
- data：包含一个或多个 32 字节非索引日志参数。
- topics：包含 0 到 4 个索引日志参数的数组，每个 32 字节。在 Solidity 中，第一个主题是事件签名的哈希值（例如 Deposit(address,bytes32,uint256)），除非您使用匿名说明符声明事件。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  -d '{"jsonrpc":"2.0","method":"eth_getLogs","params":[{"blockHash":
"0x7c5a35e9cb3e8ae0e221ab470abae9d446c3a5626ce6689fc777dcffcab52c70", "topics":
["0x241ea03ca20251805084d27d4440371c34a0b85ff108f6bb5611248f73818b80"]}],"id":74}'
```

1.5.2.29 eth_getFilterChanges

简介

过滤器的轮询方法，返回自上次轮询以来发生的日志数组。过滤器必须通过调用eth_newFilter、eth_newBlockFilter、eth_newPendingTransactionFilter来创建。该API所消耗的计算单元为20。

参数说明

参数	类型	说明
过滤器ID	String	表示过滤器 ID 的字符串。

返回值

- log object array：（数组）日志对象数组，如果自上次轮询以来没有任何更改，则为空数组。
- 对于使用eth_newBlockFilter返回值创建的过滤器，返回值是块哈希（32 字节），例如["0x3454645634534..."]。
- 对于使用eth_newFilter日志创建的过滤器，对象具有以下参数：
  - address: 该日志的来源地址。
  - blockHash:该日志所在块的哈希值。当它是待处理（Pending）日志时返回NULL。
  - blockNumber：该日志所在的块号。当它是待处理（Pending）日志时返回NULL。
  - data： 包含日志的非索引参数。
  - logIndex：块中日志索引位置的十六进制。当它是待处理（Pending）日志时返回NULL。
  - removed：若日志由于链重组而被删除，则返回true。如果它是有效的日志，则返回false。
  - topics：数据数组。索引日志参数的 0 到 4 个 32 字节 DATA 的数组。在 Solidity 中，第一个topic是事件签名的哈希值（例如 Deposit(address,bytes32,uint256)），除非您使用匿名说明符声明事件。
  - transactionHash: 32 字节。创建此日志的事务的哈希值。当它是待处理（Pending）日志时返回NULL。
  - transactionIndex：创建此日志的事务索引位置的十六进制。当它是待处理（Pending）日志时返回NULL。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
   -X POST \
   -H "Content-Type: application/json" \
   -d '{"jsonrpc":"2.0","method":"eth_getFilterChanges","params":["0x16"],"id":73}'
```

1.5.2.30 eth_getFilterLogs

简介

返回与给定过滤器 ID 匹配的所有日志的数组。该API所消耗的计算单元为75。

参数说明

参数	类型	说明
address	String	可选参数，合约地址（ 20字节 ）或日志应源自的地址列表。
fromBlock	String	可选参数，默认为“latest”，十六进制区块号，或字符串latest，earliest或pending。
toBlock	String	可选参数，默认为“latest”，十六进制区块号，或字符串latest，earliest或pending。
topics	String	可选参数，32 字节 DATA 主题数组。主题与顺序相关。
blockhash	String	可选参数，将返回的日志限制为 32 字节哈希中引用的单个块blockHash。使用blockHash相当于设置 fromBlock和 toBlock中引用的区块均为blockHash所对应的区块。如果blockHash出现在过滤条件中，则fromBlock和toBlock都不能够被设置。

返回值

- log 对象数组：与过滤器匹配的日志对象数组。对于自上次轮询以来发生的日志数组，请使用eth_getFilterChanges。日志对象包含以下键及其值：
  - address: 该日志的来源地址。
  - blockHash：该日志所在块的哈希值。当它是待处理（ Pending ）日志时返回NULL。
  - blockNumber：该日志所在的块号。当它是待处理（ Pending ）日志时返回NULL。
  - data：包含日志的非索引参数。
  - logIndex：块中日志索引位置的十六进制。当它是待处理（ Pending ）日志时返回NULL。
  - removed：若日志由于链重组而被删除，则返回true。如果它是有效的日志，则返回false。
  - topics：数据数组。索引日志参数的 0 到 4 个 32 字节 DATA 的数组。在 Solidity 中，第一个topic是事件签名的哈希值（ 例如 Deposit(address,bytes32,uint256) ），除非您使用匿名说明符声明事件。

- transactionHash: 创建此日志的事务的哈希值。当它是待处理（ Pending ）日志时返回NULL。
- transactionIndex：创建此日志的事务索引位置的十六进制。当它是待处理（ Pending ）日志时返回NULL。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
   -X POST \
   -H "Content-Type: application/json" \
   -d '{"jsonrpc":"2.0","method":"eth_getFilterLogs","params":["0x16"],"id":74}'
```

1.5.2.31 eth_newBlockFilter

简介

在节点中创建一个过滤器，以在新区块到达时发出通知。该API所消耗的计算单元为20。

参数说明

此方法不接受任何参数。

返回值

十六进制表示的新创建的过滤器ID。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
   -X POST \
   -H "Content-Type: application/json" \
   -d '{"jsonrpc":"2.0","method":"eth_newBlockFilter","params":[],"id":73}'
```

1.5.2.32 eth_newFilter

简介

根据给定的过滤器选项创建过滤器对象，以在状态更改（ 日志 ）时发出通知。该API所消耗的计算单元为20。

参数说明

参数	类型	说明
address	String	可选参数，合约地址（ 20字节 ）或日志应源自的地址列表。
fromBlock	String	可选参数，默认为“latest”，十六进制区块号，或字符串latest，earliest或pending。

参数	类型	说明
toBlock	String	可选参数，默认为“latest”，十六进制区块号，或字符串latest，earliest或pending。
topics	String	可选参数，32 字节 DATA 主题数组。主题与顺序相关。

返回值

十六进制表示的新创建的过滤器ID。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
   -X POST \
   -H "Content-Type: application/json" \
   -d '{"jsonrpc":"2.0","method":"eth_newFilter","params":[{"topics":
["0xddf252ad1be2c89b69c2b068fc378daa952ba7f163c4a11628f55a4df523b3ef"]}],"id":73}'
```

1.5.2.33 eth_newPendingTransactionFilter

简介

在节点中创建一个过滤器，以在新的待处理事务到达 Polygon 时发出通知。该API所消耗的计算单元为20。

参数说明

此方法不接受任何参数。

返回值

十六进制表示的新创建的过滤器ID。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
   -X POST \
   -H "Content-Type: application/json" \
   -d '{"jsonrpc":"2.0","method":"eth_newPendingTransactionFilter","params":[],"id":73}'
```

1.5.2.34 eth_uninstallFilter

简介

卸载具有给定 id 的过滤器。当不再需要观察时调用。此外，如果一段时间内没有通过eth_getFilterChanges请求过滤器，过滤器就会超时。该API所消耗的计算单元为17。

参数说明

参数	类型	说明
过滤器ID	String	要卸载的过滤器ID的字符串。

返回值

如果过滤器已成功卸载，返回true，否则false。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
    -X POST \
    -H "Content-Type: application/json" \
    -d '{"jsonrpc":"2.0","method":"eth_uninstallFilter","params":["0xb"],"id":73}'
```

1.5.2.35 eth_chainId

简介

返回当前配置的链 ID。该API所消耗的计算单元为1。

参数说明

此方法不接受任何参数。

返回值

十六进制表示的链 ID。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
    -X POST \
    -H "Content-Type: application/json" \
    -d '{"jsonrpc":"2.0","method":"eth_chainId","params": [],"id":1}'
```

1.5.2.36 web3_sha3

简介

返回给定数据的 Keccak-256 编码结果（ 不是标准化 SHA3-256 ）。该API所消耗的计算单元为15。

参数说明

参数	类型	说明
数据	String	需要转换的数据

返回值

给定输入的SHA3编码后的结果。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
   -X POST \
   -H "Content-Type: application/json" \
   -d '{"jsonrpc":"2.0","method":"web3_sha3","params": ["0x68656c6c6f20776f726c64"],"id":64}'
```

1.5.2.37 web3_clientVersion

简介

返回当前客户端的版本。该API所消耗的计算单元为15。

参数说明

此方法不接受任何参数。

返回值

客户端的版本。

1.5.2.38 net_version

简介

返回当前网络版本。该API所消耗的计算单元为1。

参数说明

此方法不接受任何参数。

返回值

当前网络版本

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
   -X POST \
   -H "Content-Type: application/json" \
   -d '{"jsonrpc":"2.0","method":"net_version","params": [],"id":1}'
```

1.5.2.39 net_listening

简介

返回客户端是否正在监听网络。该API所消耗的计算单元为1。

参数说明

此方法不接受任何参数。

返回值

客户端是否正在监听网络

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  -d '{"jsonrpc":"2.0","method":"net_listening","params": [], "id":1}'
```

2 波场 Tron 节点引擎

2.1 波场 Tron 介绍

Tron是一个开源同时支持智能合约功能的公共区块链平台，Tron与Ethereum是兼容的，也就是说您可以将Ethereum上智能合约直接或者稍加修改后迁移到Tron上。Tron依靠独有的共识机制使得Tron网络的TPS远远超出Ethereum，为开发者带来更加快速的交易体验。

波场官方链接：**开发者中心**，**白皮书**

用户可以通过使用华为云公链节点引擎，来提升区块链使用与开发的效率，增强其稳定性与私密性。**华为云将永远不会收集用户的区块链地址。**

说明

- **支持网络**
  - 波场主网：HTTP
  - Nile测试网：HTTP
- **波场支持API列表**

2.2 HTTP 请求示例

2.2.1 使用 cURL 发送 HTTP API 请求

Request example (With Credential):

```
curl -X GET https://your-http-endpoint/your-credential/wallet/getnowblock
```

Request example(With IAM Token):

```
curl -X GET 'X-Auth-Token:your-iam-token' https://your-http-endpoint/wallet/getnowblock
```

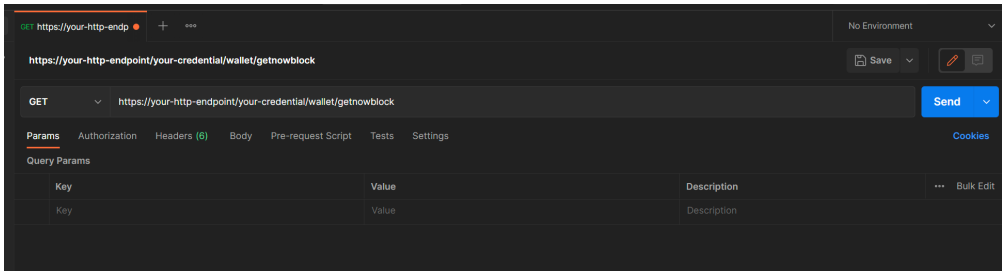
Response example:

```
{
  "blockID": "000000000204b46379ebc1d66a41a816e1b8d0c3e5f917a6af5e4471288715ef",
  "block_header": {
    "raw_data": {
      "number": 33862755,
      "txTrieRoot":
```

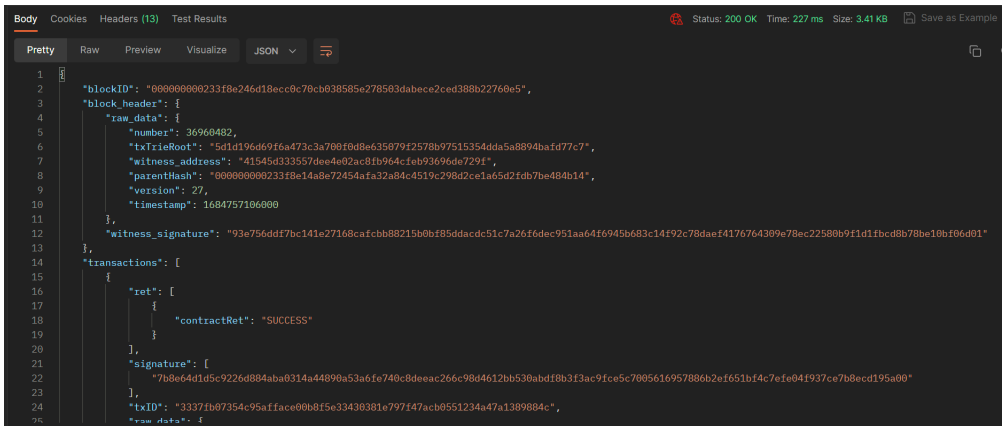
```
"0000000000000000000000000000000000000000000000000000000000000000",
    "witness_address": "TVK7n5qks4vNrkfBnjTzEs8XWF3gbv9oNr",
    "parentHash":
"000000000204b462a2825cafbae6838e8ede6e223cc3ef52ef63ee29679123b7",
    "version": 27,
    "timestamp": 1684719705000
  },
  "witness_signature":
"dbf106c634b5e7cb2710bd93dcdd38ef0be012c197866a2c3f0ceb99cbce76073cda4af405f0e1236
214bd1cd9f4a3122c42ae94e727912a847a760b6c9ca91000"
 }
}
```


2.2.2 使用 post-man 发送 HTTP API 请求

Request example:



Response example:



2.3 JSON-RPC 请求示例

2.3.1 使用 cURL 发送 JSON-RPC API 请求

Request example(With Credential):

```
curl -k -X POST https://your-http-endpoint/your-credential/jsonrpc --data
'{"jsonrpc":"2.0","method":"eth_syncing","params":[],"id":64}'
```

Request example(With IAM Token):

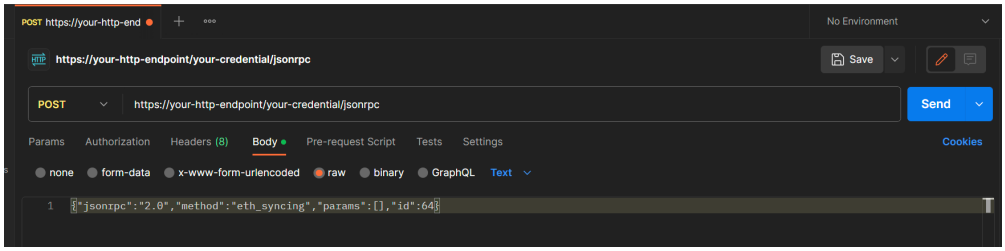
```
curl -k -X POST 'X-Auth-Token:your-iam-token' https://your-http-endpoint/jsonrpc --data
'{"jsonrpc":"2.0","method":"eth_syncing","params":[],"id":64}'
```

Response example:

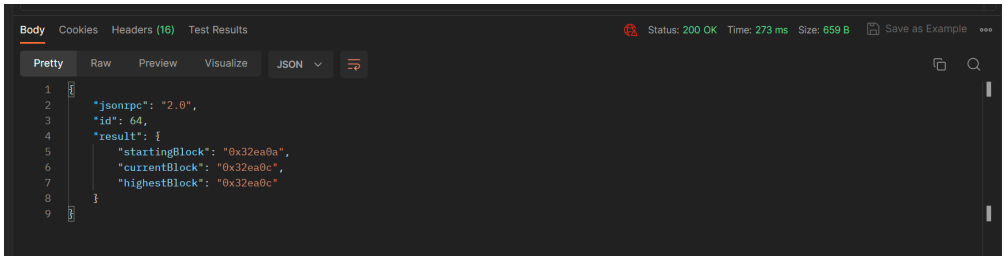
```
{
  "jsonrpc": "2.0",
  "id": 64,
  "result": {
      "startingBlock": "0x32e92c",
      "currentBlock": "0x32e92e",
      "highestBlock": "0x32e92e"
  }
}
```

2.3.2 使用 post-man 发送 JSON-RPC API 请求

Request example:



Response example:



2.4 gRPC 请求示例

2.4.1 使用 trident-sdk 发送 gRPC 请求

注意

trident sdk截止至版本0.7.0，尚未支持tls认证的方式进行gRPC请求，因此需要按照示例代码，通过java反射的方式，打开trident的tls认证进行节点对接。

在节点页面下载证书，将压缩包内的ca.crt证书放在项目内的目录。

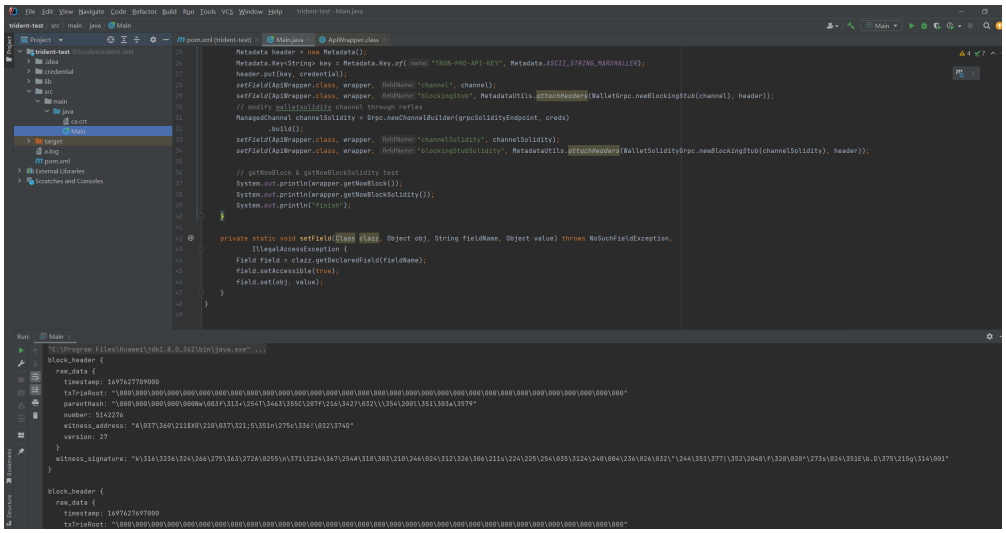
在sdk中配置gRPC endpoint，示例代码如下：

```
import io.gRPC.*;
import io.gRPC.stub.MetadataUtils;
import org.tron.trident.api.WalletgRPC;
import org.tron.trident.api.WalletSoliditygRPC;
import org.tron.trident.core.ApiWrapper;
import org.tron.trident.core.exceptions.IllegalException;
import org.tron.trident.core.key.KeyPair;

import java.io.File;
import java.io.IOException;
import java.lang.reflect.Field;

public class Main {
    public static void main(String[] args) throws IllegalException, IOException,
NoSuchFieldException,
            IllegalAccessException {
        String gRPCEndpoint = "your-gRPC-endpoint";
        String gRPCSolidityEndpoint = "your-gRPCsolidity-endpoint";
        String credential = "your-credential";
        String hexPrivateKey = "your-hex-private-key";
        ApiWrapper wrapper = new ApiWrapper(gRPCEndpoint, gRPCSolidityEndpoint,
hexPrivateKey, credential);
        // modify wallet channel through reflex
        ChannelCredentials creds = TlsChannelCredentials.newBuilder()
                .trustManager(new File("your-ca.crt-file-path"))
                .build();
        ManagedChannel channel = gRPC.newChannelBuilder(gRPCEndpoint, creds)
                .build();
        Metadata header = new Metadata();
        Metadata.Key<String> key = Metadata.Key.of("TRON-PRO-API-KEY",
Metadata.ASCII_STRING_MARSHALLER);
        header.put(key, credential);
        setField(ApiWrapper.class, wrapper, "channel", channel);
        setField(ApiWrapper.class, wrapper, "blockingStub",
MetadataUtils.attachHeaders(WalletgRPC.newBlockingStub(channel), header));
        // modify walletsolidity channel through reflex
        ManagedChannel channelSolidity = gRPC.newChannelBuilder(gRPCSolidityEndpoint, creds)
                .build();
        setField(ApiWrapper.class, wrapper, "channelSolidity", channelSolidity);
        setField(ApiWrapper.class, wrapper, "blockingStubSolidity",
MetadataUtils.attachHeaders(WalletSoliditygRPC.newBlockingStub(channelSolidity), header));
        // getNowBlock & getNowBlockSolidity test
        System.out.println(wrapper.getNowBlock());
        System.out.println(wrapper.getNowBlockSolidity());
        System.out.println("finish");
    }
    private static void setField(Class clazz, Object obj, String fieldName, Object value) throws
NoSuchFieldException,
            IllegalAccessException {
        Field field = clazz.getDeclaredField(fieldName);
        field.setAccessible(true);
        field.set(obj, value);
    }
}
```

Response example:



2.4.2 使用 gotron-sdk 发送 gRPC 请求

注意

gotron-sdk截止至版本2.3.0，默认方式创建gRPC client暂不支持gRPC solidity节点连接。

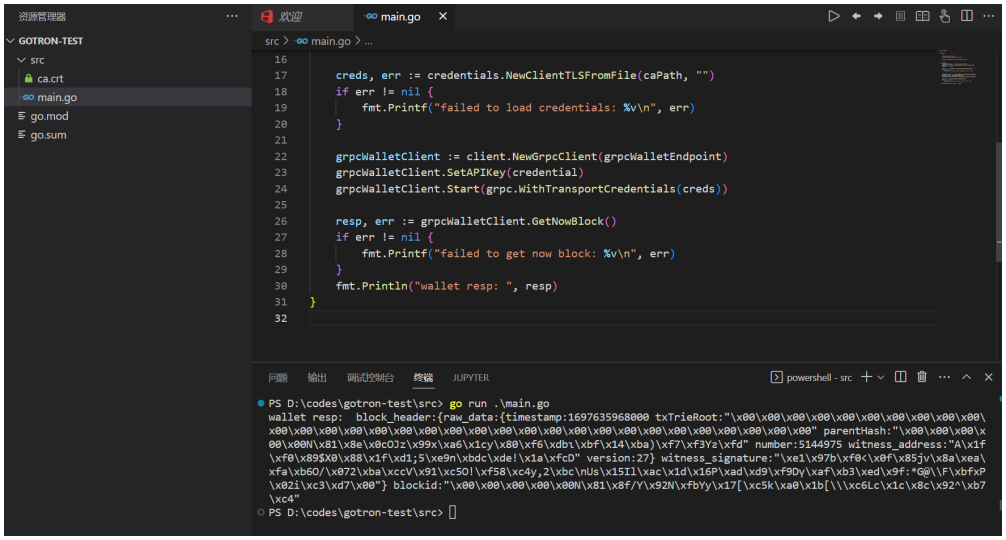
在节点页面下载证书，将压缩包内的ca.crt证书放在项目内的目录。

在sdk中配置gRPC endpoint，示例代码如下：

```
import (
    "fmt"
    "google.golang.org/gRPC"
    "google.golang.org/gRPC/credentials"
    "github.com/fbsobreira/gotron-sdk/pkg/client"
)
func main() {
    gRPCWalletEndpoint := "your-gRPC-endpoint"
    credential := "your-credential"
    caPath := "your-ca.crt-file-path"
    creds, err := credentials.NewClientTLSFromFile(caPath, "")
    if err != nil {
        fmt.Printf("failed to load credentials: %v\n", err)
    }
    gRPCWalletClient := client.NewgRPCClient(gRPCWalletEndpoint)
    gRPCWalletClient.SetAPIKey(credential)
    gRPCWalletClient.Start(gRPC.WithTransportCredentials(creds))
    resp, err := gRPCWalletClient.GetNowBlock()
```

```
    if err != nil {
        fmt.Printf("failed to get now block: %v\n", err)
    }
    fmt.Println("wallet resp: ", resp)
}
```

Response example:



2.5 应用程序开发介绍

2.5.1 使用 TronWeb 发送 HTTP 请求

Request example:

```
const TronWeb = require('tronweb');
const tronWeb = new TronWeb({
    fullHost: 'your-http-endpoint|your-credential|', // Tron HTTP终端节点
  });

let account = "accountAddress"; // 待查询的账户

const main = async () => {
    let accountBalance = await tronWeb.trx.getBalance(account);
    console.log("accountBalance:\n", accountBalance);
}

main();
```

Response example:

```
accountBalance:
 1998899400
```

2.5.2 使用 trident-sdk 发送 gRPC 请求

参考**使用trident-sdk发送gRPC请求**

2.5.3 使用 gotron-sdk 发送 gRPC 请求

参考**使用gotron-sdk发送gRPC请求**

2.6 波场 API 列表

表 2-1 可用波场 API 列表-jsonrpc

API方法	类型	说明	流控值（次/s）
eth_accounts	POST	返回客户端拥有的地址列表，tron将返回空列表。	1000
eth_blocknumber	POST	获取最新区块号。	1000
eth_call	POST	立即执行消息调用，而不在区块链上创建交易，即triggerConstantContract。	1000
eth_chainId	POST	返回TRON chainId，TRON chainId为创世块哈希的最后四个字节。	1000
eth_coinbase	POST	获取当前节点的witness address。	1000
eth_estimateGas	POST	通过triggerConstantContract预估能量消耗。	1000
eth_gasPrice	POST	获取当前的能量单价（以sun为单位）。	1000
eth_getBalance	POST	获取给定地址的账户余额。	1000
eth_getBlockByHash	POST	根据区块哈哈希获取区块信息。	50
eth_getBlockByNumber	POST	根据区块号获取区块信息。	50
eth_getBlockTransactionCountByHash	POST	根据区块哈希获取区块内的交易数量。	1000
eth_getBlockTransactionCountByNumber	POST	根据区块号获取区块内的交易数量。	1000
eth_getCode	POST	获取给定智能合约的runtime code。	400
eth_getStorageAt	POST	返回某地址的指定位置存储的内容，可用于获取某个合约中某个变量的值。	1000
eth_getTransactionByBlockHashAndIndex	POST	根据区块哈希，获取区块的第index个交易。	1000

API方法	类型	说明	流控值（次/s）
eth_getTransactionByBlockNumberAndIndex	POST	根据区块号，获取区块的第index个交易。	1000
eth_getTransactionByHash	POST	根据交易哈希获取交易信息。	1000
eth_getTransactionReceipt	POST	查询交易的 Info 信息，包括交易的 fee 信息、所在区块和虚拟机log等。	1000
eth_getWork	POST	获取当前区块的哈希。	1000
eth_protocolVersion	POST	获取java-tron block版本。	1000
eth_syncing	POST	获取节点的同步状态。	1000
eth_newFilter	POST	创建一个事件过滤器对象，监听事件。	1000
eth_newBlockFilter	POST	创建一个过滤器，当有新块到达时获得通知。	1000
eth_getFilterChanges	POST	返回自上次轮询以来发生的事件列表。	1000
eth_getFilterLogs	POST	返回符合过滤条件的所有事件。	10
eth_uninstallFilter	POST	取消一个过滤器，当不再需要监控时需取消过滤器。	1000
eth_getLogs	POST	返回与给定过滤条件匹配的所有事件。	10
net_listening	POST	查询客户端是否处于监听网络连接的状态。	1000
net_peerCount	POST	返回当前节点所连接的peer节点数量。	1000
net_version	POST	返回创世块的哈希值。	1000
web3_clientVersion	POST	返回当前节点的版本。	1000
web3_sha3	POST	计算给定数据的Keccak-256值（不是标准的SHA3-256）。	1000

API方法	类型	说明	流控值（次/s）
buildTransaction	POST	创建transaction，交易类型不同，参数不同。	1000

表 2-2 可用波场 API 列表-gRPC

API方法	说明	流控值（次/s）
BroadcastTransaction	广播签名后的交易。	1000
CreatetTansaction	创建交易。（ 请使用CreateTransaction2 ）	1000
CreateTransaction2	创建交易。	1000
CreateAccount	激活一个链上账户。（ 请使用CreateAccount2 ）	1000
CreateAccount2	激活一个链上账户。	1000
GetAccount	查询一个账号的信息，包括TRX余额、TRC10余额、质押以获取资源情况、投票情况以及权限等。	600
UpdateAccount	修改账户的名称。(请使用UpdateAccount2)	1000
UpdateAccount2	修改账户的名称。	1000
VoteWitnessAccount	对超级代表进行投票，返回投票的 Transaction，需要签名后广播。（ 请使用 VoteWitnessAccount2 ）	1000
VoteWitnessAccount2	对超级代表进行投票，返回投票的 Transaction，需要签名后广播。	1000
UpdateSetting	更新合约的 consume_user_resource_percent 配置，返回未签名交易，需要签名后广播。	1000

API方法	说明	流控值（次/s）
UpdatEenergyLimit	更新合约的 origin_energy_limit，返回未签名交易，需要签名后广播。	1000
CreateAssetIssue	发行TRC10通证。（请使用CreateAssetIssue2）	1000
CreateAssetIssue2	发行TRC10通证。	1000
UpdateWitness	修改witness配置信息中的 URL，需要签名后广播。（请使用UpdateWitness2）	1000
UpdateWitness2	修改witness配置信息中的 URL，需要签名后广播。	1000
CreateWitness	申请成为超级代表， 返回申请超级代表的Transaction，需要签名后广播。（请使用CreateWitness2）	1000
CreateWitness2	申请成为超级代表， 返回申请超级代表的Transaction，需要签名后广播。	1000
TransferAsset	转账TRC10通证。（请使用TransferAsset2）	1000
TransferAsset2	转账TRC10通证。	1000
ParticipateAssetIssue	参与TRC10通证发行。（请使用ParticipateAssetIssue2）	1000
ParticipateAssetIssue2	参与TRC10通证发行。	1000
FreezeBalance2	在Stake2.0质押机制下，质押TRX以获取带宽或者能量，同时根据质押额度获得等值投票权(TP)。（已废弃）	1000
FreezeBalanceV2	在Stake2.0质押机制下，质押TRX以获取带宽或者能量，同时根据质押额度获得等值投票权(TP)。	1000
UnfreezeBalance	解锁在Stake1.0期间质押的TRX，释放所得到的带宽或能量以及TP，同时会自动取消所有投票。（请使用UnfreezeBalance2）	1000
UnfreezeBalance2	解锁在Stake1.0期间质押的TRX，释放所得到的带宽或能量以及TP，同时会自动取消所有投票。	1000

API方法	说明	流控值（次/s）
UnfreezeBalanceV2	解锁通过Stake2.0机制质押的TRX，释放所相应数量的带宽和能量，同时回收相应数量的投票权(TP)。	1000
UnfreezeAsset	解锁已经结束质押期的 TRC10通证。（请使用UnfreezeAsset2）	1000
UnfreezeAsset2	解锁已经结束质押期的 TRC10通证。	1000
WithdrawBalance	超级代表或用户提取奖励，每 24 小时可调用一次。超级代表将 allowance 中的余额提取到账户中，用户将投票奖励提取到自己的账户中。（请使用WithdrawBalance2）	1000
WithdrawBalance2	超级代表或用户提取奖励，每 24 小时可调用一次。超级代表将 allowance 中的余额提取到账户中，用户将投票奖励提取到自己的账户中。	1000
WithdrawExpireUnfreeze	提取已过锁定期的解质押的本金。	1000
DelegateResource	在Stake 2.0 机制下，将带宽或者能量资源代理给其它账户。	1000
CancelAllUnfreezeV2	取消所有未完成的解质押，将过期的解质押金额提取到账户余额中，将未过期的解质押金额重新质押。	1000
UpdateAsset	修改TRC10通证基本信息。（请使用UpdateAsset2）	1000
UpdateAsset2	修改TRC10通证基本信息。	1000
ProposalCreate	创建提案交易，需要签名后广播。	1000
ProposalApprove	批准提案，需要签名后广播。	1000
ProposalDelete	删除提案，需要签名后广播。	1000
ExchangeCreate	创建交易对，需要签名后广播。警告：成功执行，签署和广播此 API 调用将从用户的帐户中扣除 1024 TRX。	1000
ExchangeInject	给交易对注资，注资后可以防止交易对价格波动太大，需要签名后广播。	1000

API方法	说明	流控值（次/s）
ExchangeWithdraw	对交易对撤资，需要签名后广播。	1000
ExchangeTransaction	参与交易对交易，需要签名后广播。	1000
GetAssetIssueByAccount	查询账户发行的TRC10通证。	1000
GetAccountNet	查询账户带宽信息。	1000
GetAccountResource	查询账户的资源信息（带宽、能量）。	1000
GetAssetIssueByName	根据通证名称查询TRC10通证。	200
GetAssetIssueListByName	根据名称返回同名的所有TRC 10代币列表。	200
GetAssetIssueById	根据ID查询TRC10通证。	1000
GetNowBlock	查询最新块。（请使用GetNowBlock2）	1000
GetNowBlock2	查询最新块。	1000
GetBlockByNum	通过高度查询区块内容。（请使用GetBlockByNum2）	15
GetBlockByNum2	通过高度查询区块内容。	15
GetTransactionCountByBlockNum	获取指定块中的交易计数。	1000
GetBlockById	通过区块ID（即区块哈希）查询区块。	15
GetBlockByLimitNext	查询指定范围的区块。（请使用GetBlockByLimitNext2）	10
GetBlockByLimitNext2	查询指定范围的区块。	10
GetBlockByLatestNum	查询最新的若干个区块。（请使用GetBlockByLatestNum2）	10
GetBlockByLatestNum2	查询最新的若干个区块。	10

API方法	说明	流控值（次/s）
GetTransactionById	按交易哈希查询交易。	1000
DeployContract	部署合约，返回 Transaction Extention，其中包含未签名的交易。	1000
GetContract	查询链上的合约信息，包括合约的bytecode、ABI、配置参数等。	300
GetContractInfo	查询链上的合约信息。与wallet/getcontract接口不同，该接口不仅返回bytecode还会返回合约的runtime bytecode。runtime bytecode相比bytecode，不包含构造函数以及构造函数的参数信息。	200
TriggerContract	调用智能合约，返回 TransactionExtention，需要签名后广播。	1000
TriggerConstantContract	调用合约只读函数，也可以调用合约非只读函数，用于预判交易是否可以执行成功或者预估交易的能量消耗，也可以预估合约部署消耗的能量。	1000
EstimateEnergy	预估智能合约调用交易或部署交易执行成功需要提供的能量。	1000
ClearContractAbi	将合约的 ABI 设置为空。返回未签名交易，需要签名后广播。	1000
ListWitnesses	返回所有超级代表的列表。	250
GetDelegatedResource	查看在stake1.0阶段一个账户代理给另外一个账户的资源情况。（请使用GetDelegatedResourceV2）	1000
GetDelegatedResourceV2	查看在stake1.0阶段一个账户代理给另外一个账户的资源情况。	1000
GetDelegatedResourceAccountIndex	查看在stake1.0阶段一个账户给哪些账户代理了资源，以及哪些账户为该账户代理了资源。	1000
GetDelegatedResourceAccountIndexV2	查询在Stake2.0阶段，某地址的资源委托索引账号。	1000
GetCanDelegatedMaxSize	查询目标地址中指定类型资源的可代理数量，单位为sun。	1000
GetAvailableUnfreezeCount	查询Stake2.0机制下，当前解质押剩余次数。	1000

API方法	说明	流控值（次/s）
GetCanWithdrawUnfreezeAmount	查询在某时间点可以提取的解质押本金数量。	1000
ListProposals	查询所有提案并返回提案信息。	300
GetProposalById	根据ID查询提案并返回提案详细信息。	1000
ListExchanges	查询所有交易对。	400
GetExchangeById	根据id查询交易对。	1000
GetChainParameters	查询当前所有提案参数，超级代表可以提议修改这些参数。	1000
GetAssetIssueList	查询所有TRC10通证列表。	5
GetPaginatedAssetIssueList	分页查询TRC10通证列表。	20
GetNextMaintenanceTime	返回下个计票时间点的时间戳（毫秒）。	1000
GetTransactionInfoById	查询交易的 Info 信息，包括交易的 fee 信息、所在区块和虚拟机log等。	1000
AccountPermissionUpdate	修改账户权限。	1000
GetTransactionSignWeight	查询交易签权重。	1000
GetTransactionApprovedList	根据交易内容和签名信息计算得到为交易签名的账户地址列表，可用交易验签。	1000
GetNodeInfo	查询当前节点的信息。	700
GetRewardInfo	查询用户未被提取的投票奖励。	1000
GetBrokerageInfo	查询超级代表佣金比例。	1000
UpdateBrokerage	更新 SR 佣金比例，需要签名后广播。	1000
GetTransactionInfoByBlockNum	获取特定区块的所有交易 Info 信息。	150

API方法	说明	流控值（次/s）
GetBurnTrx	查询自从第54号委员会提议生效后，因链上交易手续费而销毁的TRX数量。	1000
GetTransactionFromPending	从pending pool中获取交易详细信息。	1000
GetTransactionListFromPending	获取pending pool中交易列表信息。	1000
GetPendingSize	获取pending pool队列的大小。	1000
GetBlock	根据区块高度或者区块哈希查询区块头信息或者整个区块信息。	50
UnDelegateResource	在Stake 2.0机制下，取消为目标地址代理的带宽或者能量。	1000

表 2-3 可用波场 API 列表-gRPC solidity

API方法	说明	流控值（次/s）
GetAccount	查询一个账号的信息，包括TRX余额、TRC10余额、质押以获取资源情况、投票情况以及权限等。	600
ListWitnesses	返回所有超级代表的列表。	250
GetAssetIssueList	查询所有TRC10通证列表。	5
GetPaginatedAssetIssueList	分页查询TRC10通证列表。	20
GetAssetIssueByName	根据通证名称查询TRC10通证。	200
GetAssetIssueListByName	根据名称返回同名的所有TRC 10代币列表。	200
GetAssetIssueById	根据ID查询TRC10通证。	1000
GetNowBlock	查询最新块。（请使用GetNowBlock2）	1000

API方法	说明	流控值（次/s）
GetNowBlock2	查询最新块。	1000
GetBlockByNum	通过高度查询区块内容。（请使用GetBlockByNum2）	15
GetBlockByNum2	通过高度查询区块内容。	15
GetTransactionCountByBlockNum	获取指定块中的交易计数。	1000
GetDelegatedResource	查看在stake1.0阶段一个账户代理给另外一个账户的资源情况。（请使用GetDelegatedResourceV2）	1000
GetDelegatedResourceV2	查看在stake1.0阶段一个账户代理给另外一个账户的资源情况。	1000
GetDelegatedResourceAccountIndex	查看在stake1.0阶段一个账户给哪些账户代理了资源，以及哪些账户为该账户代理了资源。	1000
GetDelegatedResourceAccountIndexV2	查询在Stake2.0阶段，某地址的资源委托索引。	1000
GetCanDelegatedMaxSize	查询目标地址中指定类型资源的可代理数量，单位为sun。	1000
GetAvailableUnfreezeCount	查询Stake2.0机制下，当前解质押剩余次数。	1000
GetCanWithdrawUnfreezeAmount	查询在某时间点可以提取的解质押本金数量。	1000
GetExchangeById	根据id查询交易对。	1000
ListExchanges	查询所有交易对。	400
GetTransactionById	按交易哈希查询交易。	1000
GetTransactionInfoById	查询交易的 Info 信息，包括交易的 fee 信息、所在区块和虚拟机log等。	1000
GetRewardInfo	查询用户未被提取的投票奖励。	1000
GetBrokerageInfo	查询超级代表佣金比例。	1000

API方法	说明	流控值（次/s）
TriggerConstantContract	调用合约只读函数，也可以调用合约非只读函数，用于预判交易是否可以执行成功或者预估交易的能量消耗，也可以预估合约部署消耗的能量。	1000
EstimateEnergy	预估智能合约调用交易或部署交易执行成功需要提供的能量。	1000
GetTransactionInfoByBlockNum	获取特定区块的所有交易 Info 信息。	150
GetBurnTrx	查询自从第54号委员会提议生效后，因链上交易手续费而销毁的TRX数量。	1000
GetBlock	根据区块高度或者区块哈希查询区块头信息或者整个区块信息。	50

2.6.1 专享版

表 2-4 可用波场 API 列表-wallet

API方法	类型	说明	流控值（次/s）
/wallet/validateaddress	POST	检查地址是否格式正确。	1000
/wallet/broadcasttransaction	POST	广播签名后的交易。	1000
/wallet/broadcasthex	POST	直接广播 protobuf 序列化后的十六进制 Transaction。	1000
/wallet/createtransaction	POST	创建交易。	1000
/wallet/createaccount	POST	激活一个链上账户。	1000
/wallet/getaccount	POST	查询一个账号的信息，包括TRX余额、TRC10余额、质押以获取资源情况、投票情况以及权限等。	600

API方法	类型	说明	流控值（次/s）
/wallet/updateaccount	POST	修改账户的名称。	1000
/wallet/accountpermissionupdate	POST	修改账户权限。	1000
/wallet/getaccountresource	POST	查询账户的资源信息（带宽、能量）。	1000
/wallet/getaccountnet	POST	查询账户带宽信息。	1000
/wallet/unfreezebalance	POST	解锁在Stake1.0期间质押的TRX，释放所得到的带宽或能量以及TP，同时会自动取消所有投票。	1000
/wallet/getdelegatedresource	POST	查看在stake1.0阶段一个账户代理给另外一个账户的资源情况。	1000
/wallet/getdelegatedresourceaccountindex	POST	查看在stake1.0阶段一个账户给哪些账户代理了资源，以及哪些账户为该账户代理了资源。	1000
/wallet/freezebalancev2	POST	在Stake2.0质押机制下，质押TRX以获取带宽或者能量，同时根据质押额度获得等值投票权(TP)。	1000
/wallet/unfreezebalancev2	POST	解锁通过Stake2.0机制质押的TRX，释放所相应数量的带宽和能量，同时回收相应数量的投票权(TP)。	1000
/wallet/cancelallunfreezev2	POST	取消所有未完成的解质押，将过期的解质押金额提取到账户余额中，将未过期的解质押金额重新质押。	1000
/wallet/delegateresource	POST	在Stake 2.0 机制下，将带宽或者能量资源代理给其它账户。	1000
/wallet/undelegateresource	POST	在Stake 2.0机制下，取消为目标地址代理的带宽或者能量。	1000
/wallet/withdrawexpireunfreeze	POST	提取已过锁定期的解质押的本金。	1000
/wallet/getavailableunfreezecount	POST	查询Stake2.0机制下，当前解质押剩余次数。	1000

API方法	类型	说明	流控值（次/s）
/wallet/getcanwithdrawunfreezeamount	POST	查询在某时间点可以提取的解质押本金数量。	1000
/wallet/getcandelegatedmaxsize	POST	查询目标地址中指定类型资源的可代理数量，单位为sun。	1000
/wallet/getdelegatedresourcev2	POST	查询在Stake2.0机制下，某地址代理给目标地址的资源情况。	1000
/wallet/getdelegatedresourceaccountindexv2	POST	查询在Stake2.0阶段，某地址的资源委托索引账号。	1000
/wallet/getblock	POST	根据区块高度或者区块哈希查询区块头信息或者整个区块信息。	50
/wallet/getblockbynum	POST	通过高度查询区块内容。	15
/wallet/getblockbyid	POST	通过区块ID（即区块哈希）查询区块。	15
/wallet/getblockbylatestnum	POST	查询最新的若干个区块。	10
/wallet/getblockbylimitnext	POST	查询指定范围的区块。	10
/wallet/getnowblock	GET	查询最新块。	1000
/wallet/gettransactionbyid	POST	按交易哈希查询交易。	1000
/wallet/gettransactioninfobyid	POST	查询交易的 Info 信息，包括交易的 fee 信息、所在区块和虚拟机log等。	1000
/wallet/gettransactioninfobyblocknum	POST	获取特定区块的所有交易 Info 信息。	150
/wallet/getnodeinfo	GET	查询当前节点的信息。	700
/wallet/getchainparameters	GET	查询当前所有提案参数，超级代表可以提议修改这些参数。	1000

API方法	类型	说明	流控值（次/s）
/wallet/getenergyprices	GET	查询历史能量单价。	1000
/wallet/getbandwidthprices	GET	查询历史带宽单价。	1000
/wallet/getburntrx	GET	查询自从第54号委员会提议生效后，因链上交易手续费而销毁的TRX数量。	1000
/wallet/getapprovedlist	POST	根据交易内容和签名信息计算得到为交易签名的账户地址列表，可用交易验签。	1000
/wallet/getassetissuebyaccount	POST	查询账户发行的TRC10通证。	1000
/wallet/getassetissuebyid	POST	根据ID查询TRC10通证。	1000
/wallet/getassetissuebyname	POST	根据通证名称查询TRC10通证。	150
/wallet/getassetissuelist	GET	查询所有TRC10通证列表。	5
/wallet/getassetissuelistbyname	POST	根据名称返回同名的所有TRC 10代币列表。	150
/wallet/getpaginatedassetissuelist	POST	分页查询TRC10通证列表。	20
/wallet/transferasset	POST	转账TRC10通证。	1000
/wallet/createassetissue	POST	发行TRC10通证。	1000
/wallet/participateassetissue	POST	参与TRC10通证发行。	1000
/wallet/unfreezeasset	POST	解锁已经结束质押期的 TRC10通证。	1000
/wallet/updateasset	POST	修改TRC10通证基本信息。	1000
/wallet/getcontract	POST	查询链上的合约信息，包括合约的bytecode、 ABI、配置参数等。	300

API方法	类型	说明	流控值（次/s）
/wallet/getcontractinfo	POST	查询链上的合约信息。与wallet/getcontract接口不同，该接口不仅返回bytecode还会返回合约的runtime bytecode。runtime bytecode相比bytecode，不包含构造函数以及构造函数的参数信息。	200
/wallet/triggersmartcontract	POST	调用智能合约，返回TransactionExtention，需要签名后广播。	1000
/wallet/triggerconstantcontract	POST	调用合约只读函数，也可以调用合约非只读函数，用于预判交易是否可以执行成功或者预估交易的能量消耗，也可以预估合约部署消耗的能量。	1000
/wallet/deploycontract	POST	部署合约，返回 Transaction Extention，其中包含未签名的交易。	1000
/wallet/updatesetting	POST	更新合约的consume_user_resource_percent 配置，返回未签名交易，需要签名后广播。	1000
/wallet/updateenergylimit	POST	更新合约的 origin_energy_limit，返回未签名交易，需要签名后广播。	1000
/wallet/clearabi	POST	将合约的 ABI 设置为空。返回未签名交易，需要签名后广播。	1000
/wallet/estimateenergy	POST	预估智能合约调用交易或部署交易执行成功需要提供的能量。	1000
/wallet/createwitness	POST	申请成为超级代表， 返回申请超级代表的Transaction，需要签名后广播。	1000
/wallet/updatewitness	POST	修改witness配置信息中的 URL，需要签名后广播。	1000
/wallet/listwitnesses	GET	返回所有超级代表的列表。	200
/wallet/votewitnessaccount	POST	对超级代表进行投票，返回投票的Transaction， 需要签名后广播。	1000
/wallet/updateBrokerage	POST	更新 SR 佣金比例，需要签名后广播。	1000
/wallet/getBrokerage	POST	查询超级代表佣金比例。	1000
/wallet/getReward	POST	查询用户未被提取的投票奖励。	1000

API方法	类型	说明	流控值（次/s）
/wallet/withdrawbalance	POST	超级代表或用户提取奖励，每 24 小时可调用一次。超级代表将 allowance 中的余额提取到账户中，用户将投票奖励提取到自己的账户中。	1000
/wallet/getnextmaintenancetime	GET	返回下个计票时间点的时间戳（毫秒）。	1000
/wallet/proposalcreate	POST	创建提案交易，需要签名后广播。	1000
/wallet/proposalapprove	POST	批准提案，需要签名后广播。	1000
/wallet/proposaldelete	POST	删除提案，需要签名后广播。	1000
/wallet/listproposals	GET	查询所有提案并返回提案信息。	300
/wallet/getproposalbyid	POST	根据ID查询提案并返回提案详细信息。	1000
/wallet/exchangecreate	POST	创建交易对，需要签名后广播。警告：成功执行，签署和广播此 API 调用将从用户的帐户中扣除 1024 TRX。	1000
/wallet/exchangeinject	POST	给交易对注资，注资后可以防止交易对价格波动太大，需要签名后广播。	1000
/wallet/exchangewithdraw	POST	对交易对撤资，需要签名后广播。	1000
/wallet/exchangetransaction	POST	参与交易对交易，需要签名后广播。	1000
/wallet/getexchangebyid	POST	根据id查询交易对。	1000
/wallet/listexchanges	GET	查询所有交易对。	400
/wallet/gettransactionlistfrompending	GET	获取pending pool中交易列表信息。	1000
/wallet/gettransactionfrompending	POST	从pending pool中获取交易详细信息。	1000

API方法	类型	说明	流控值（次/s）
/wallet/getpendingsize	GET	获取pending pool队列的大小。	1000
/wallet/getsignweight	POST	查询交易签名权重。	1000

表 2-5 可用波场 API 列表-walletsolidity

API方法	类型	说明	流控值（次/s）
/walletsolidity/gettransactionbyid	POST	按交易哈希查询交易（已固化状态）。	1000
/walletsolidity/gettransactioninfobyid	POST	查询交易的 Info 信息，包括交易的 fee 信息、 所在区块、虚拟机log等（已固化状态）。	1000
/walletsolidity/gettransactioninfobyblocknum	POST	获取特定区块的所有交易 Info 信息（已固化状态）。	1000
/walletsolidity/gettransactioncountbyblocknum	POST	按区块号查询区块内交易数量（已固化状态）。	1000
/walletsolidity/getblock	POST	根据区块高度或者区块哈希查询区块头信息或者整个区块信息（已固化状态）。	1000
/walletsolidity/getblockbyid	POST	通过区块ID（即区块哈希）查询区块（已固化状态）。	1000
/walletsolidity/getblockbylatestnum	POST	查询最新的若干个区块（已固化状态）。	1000
/walletsolidity/getblockbylimitnext	POST	查询指定范围的区块（已固化状态）。	1000
/walletsolidity/getblockbynum	POST	查询确认指定块是否被固化。	1000
/walletsolidity/getnowblock	GET	查询当前最新区块（已固化状态）。	1000

API方法	类型	说明	流控值（次/s）
/walletsolidity/getaccount	POST	获取账户信息（已固化状态）。	1000
/walletsolidity/getdelegatedresource	POST	查看stake1.0阶段一个账户代理给另外一个账户的资源情况（已固化状态）。	1000
/walletsolidity/getdelegatedresourcev2	POST	Stake 2.0 API：查询某地址代理给目标地址的资源情况（已固化状态）。	1000
/walletsolidity/getcandelegatedmaxsize	POST	Stake 2.0 API: 查询目标地址中指定类型资源的可代理数量（已固化状态），单位为sun。	1000
/walletsolidity/getavailableunfreezecount	POST	Stake 2.0 API: 查询当下解质押剩余次数（已固化状态）。	1000
/walletsolidity/getcanwithdrawunfreezeamount	POST	Stake 2.0 API: 查询在某时间点可以提取的解质押本金数量（已固化状态）。	1000
/walletsolidity/getdelegatedresourceaccountindexv2	POST	Stake 2.0 API：查询某地址的资源委托索引（已固化状态）。返回两个列表，一个是该帐户将资源委托给的地址列表（toAddress）；另一个是将资源委托给该帐户的地址列表（fromAddress）。	1000
/walletsolidity/getnodeinfo	GET	查询当前节点的信息（已固化状态）。	1000
/walletsolidity/getburntrx	GET	查询自从第54号委员会提议生效后，因链上交易手续费而销毁的TRX数量（已固化状态）。	1000
/walletsolidity/triggerconstantcontract	POST	既可以调用合约只读函数（view 或 pure修饰的函数），用于查询合约已固化状态数据，也可以调用合约非只读函数，用于在已固化状态下预判交易是否可以执行成功或者预估交易的能量消耗。	1000
/walletsolidity/estimateenergy	POST	在已固化状态下，预估智能合约交易执行成功需要提供的能量。	1000
/walletsolidity/getassetissuebyid	POST	根据ID查询TRC10通证（已固化状态）。	1000
/walletsolidity/getassetissuebyname	POST	根据通证名称查询TRC10通证（已固化状态）。	150

API方法	类型	说明	流控值（次/s）
/walletsolidity/getassetissuelist	GET	查询所有TRC10通证列表（已固化状态）。	5
/walletsolidity/getassetissuelistbyname	POST	根据名称返回同名的所有TRC10代币列表（已固化状态）。	150
/walletsolidity/getpaginatedassetissuelist	POST	分页查询TRC10通证列表（已固化状态）。	20
/walletsolidity/listwitnesses	GET	返回所有超级代表的列表（已固化状态）。	20
/walletsolidity/getBrokerage	POST	查询超级代表佣金比例（已固化状态）。	1000
/walletsolidity/getReward	POST	查询用户未被提取的投票奖励（已固化状态）。	1000
/walletsolidity/getexchangebyid	POST	根据id查询交易对（已固化状态）。	1000
/walletsolidity/listexchanges	GET	查询所有交易对（已固化状态）。	1000
/walletsolidity/getenergyprices	GET	查询历史能量单价。	1000
/walletsolidity/getbandwidthprices	GET	查询历史带宽单价。	1000

表 2-6 可用波场 API 列表-jsonrpc

API方法	类型	说明	流控值（次/s）
eth_accounts	POST	返回客户端拥有的地址列表，tron将返回空列表。	1000
eth_blocknumber	POST	获取最新区块号。	1000

API方法	类型	说明	流控值（次/s）
eth_call	POST	立即执行消息调用，而不在区块链上创建交易，即triggerConstantContract。	1000
eth_chainId	POST	返回TRON chainId，TRON chainId为创世块哈希的最后四个字节。	1000
eth_coinbase	POST	获取当前节点的witness address。	1000
eth_estimateGas	POST	通过triggerConstantContract预估能量消耗。	1000
eth_gasPrice	POST	获取当前的能量单价（以sun为单位）。	1000
eth_getBalance	POST	获取给定地址的账户余额。	1000
eth_getBlockByHash	POST	根据区块哈哈希获取区块信息。	50
eth_getBlockByNumber	POST	根据区块号获取区块信息。	50
eth_getBlockTransactionCountByHash	POST	根据区块哈希获取区块内的交易数量。	1000
eth_getBlockTransactionCountByNumber	POST	根据区块号获取区块内的交易数量。	1000
eth_getCode	POST	获取给定智能合约的runtime code。	400
eth_getStorageAt	POST	返回某地址的指定位置存储的内容，可用于获取某个合约中某个变量的值。	1000
eth_getTransactionByBlockHashAndIndex	POST	根据区块哈希，获取区块的第index个交易。	1000
eth_getTransactionByBlockNumberAndIndex	POST	根据区块号，获取区块的第index个交易。	1000
eth_getTransactionByHash	POST	根据交易哈希获取交易信息。	1000
eth_getTransactionReceipt	POST	查询交易的 Info 信息，包括交易的 fee 信息、所在区块和虚拟机log等。	1000
eth_getWork	POST	获取当前区块的哈希。	1000

API方法	类型	说明	流控值（次/s）
eth_protocolVersion	POST	获取java-tron block版本。	1000
eth_syncing	POST	获取节点的同步状态。	1000
eth_newFilter	POST	创建一个事件过滤器对象，监听事件。	1000
eth_newBlockFilter	POST	创建一个过滤器，当有新块到达时获得通知。	1000
eth_getFilterChanges	POST	返回自上次轮询以来发生的事件列表。	1000
eth_getFilterLogs	POST	返回符合过滤条件的所有事件。	10
eth_uninstallFilter	POST	取消一个过滤器，当不再需要监控时需取消过滤器。	1000
eth_getLogs	POST	返回与给定过滤条件匹配的所有事件。	10
net_listening	POST	查询客户端是否处于监听网络连接的状态。	1000
net_version	POST	返回创世块的哈希值。	1000
web3_clientVersion	POST	返回当前节点的版本。	1000
web3_sha3	POST	计算给定数据的Keccak-256值（不是标准的SHA3-256）。	1000
buildTransaction	POST	创建transaction，交易类型不同，参数不同。	1000

表 2-7 可用波场 API 列表-gRPC

API方法	说明	流控值（次/s）
BroadcastTransaction	广播签名后的交易。	1000
CreatetTansaction	创建交易。（请使用CreateTransaction2）	1000
CreateTransaction2	创建交易。	1000
CreateAccount	激活一个链上账户。（请使用CreateAccount2）	1000
CreateAccount2	激活一个链上账户。	1000
GetAccount	查询一个账号的信息，包括TRX余额、TRC10余额、质押以获取资源情况、投票情况以及权限等。	600
UpdateAccount	修改账户的名称。(请使用UpdateAccount2)	1000
UpdateAccount2	修改账户的名称。	1000
VoteWitnessAccount	对超级代表进行投票，返回投票的 Transaction，需要签名后广播。（请使用VoteWitnessAccount2）	1000
VoteWitnessAccount2	对超级代表进行投票，返回投票的 Transaction，需要签名后广播。	1000
UpdateSetting	更新合约的 consume_user_resource_percent 配置，返回未签名交易，需要签名后广播。	1000
UpdatEenergyLimit	更新合约的 origin_energy_limit，返回未签名交易，需要签名后广播。	1000
CreateAssetIssue	发行TRC10通证。（请使用CreateAssetIssue2）	1000
CreateAssetIssue2	发行TRC10通证。	1000
UpdateWitness	修改witness配置信息中的 URL，需要签名后广播。（请使用UpdateWitness2）	1000
UpdateWitness2	修改witness配置信息中的 URL，需要签名后广播。	1000

API方法	说明	流控值（次/s）
CreateWitness	申请成为超级代表， 返回申请超级代表的Transaction，需要签名后广播。（ 请使用CreateWitness2 ）	1000
CreateWitness2	申请成为超级代表， 返回申请超级代表的Transaction，需要签名后广播。	1000
TransferAsset	转账TRC10通证。（ 请使用TransferAsset2 ）	1000
TransferAsset2	转账TRC10通证。	1000
ParticipateAssetIssue	参与TRC10通证发行。（ 请使用ParticipateAssetIssue2 ）	1000
ParticipateAssetIssue2	参与TRC10通证发行。	1000
FreezeBalance2	在Stake2.0质押机制下，质押TRX以获取带宽或者能量，同时根据质押额度获得等值投票权(TP)。（ 已废弃 ）	1000
FreezeBalanceV2	在Stake2.0质押机制下，质押TRX以获取带宽或者能量，同时根据质押额度获得等值投票权(TP)。	1000
UnfreezeBalance	解锁在Stake1.0期间质押的TRX，释放所得到的带宽或能量以及TP，同时会自动取消所有投票。（ 请使用UnfreezeBalance2 ）	1000
UnfreezeBalance2	解锁在Stake1.0期间质押的TRX，释放所得到的带宽或能量以及TP，同时会自动取消所有投票。	1000
UnfreezeBalanceV2	解锁通过Stake2.0机制质押的TRX，释放所相应数量的带宽和能量，同时回收相应数量的投票权(TP)。	1000
UnfreezeAsset	解锁已经结束质押期的 TRC10通证。（ 请使用UnfreezeAsset2 ）	1000
UnfreezeAsset2	解锁已经结束质押期的 TRC10通证。	1000
WithdrawBalance	超级代表或用户提取奖励，每 24 小时可调用一次。超级代表将 allowance 中的余额提取到账户中，用户将投票奖励提取到自己的账户中。（ 请使用WithdrawBalance2 ）	1000

API方法	说明	流控值（次/s）
WithdrawBalance2	超级代表或用户提取奖励，每 24 小时可调用一次。超级代表将 allowance 中的余额提取到账户中，用户将投票奖励提取到自己的账户中。	1000
WithdrawExpireUnfreeze	提取已过锁定期的解质押的本金。	1000
DelegateResource	在Stake 2.0 机制下，将带宽或者能量资源代理给其它账户。	1000
CancelAllUnfreezeV2	取消所有未完成的解质押，将过期的解质押金额提取到账户余额中，将未过期的解质押金额重新质押。	1000
UpdateAsset	修改TRC10通证基本信息。（请使用 UpdateAsset2）	1000
UpdateAsset2	修改TRC10通证基本信息。	1000
ProposalCreate	创建提案交易，需要签名后广播。	1000
ProposalApprove	批准提案，需要签名后广播。	1000
ProposalDelete	删除提案，需要签名后广播。	1000
ExchangeCreate	创建交易对，需要签名后广播。警告：成功执行，签署和广播此 API 调用将从用户的帐户中扣除 1024 TRX。	1000
ExchangeInject	给交易对注资，注资后可以防止交易对价格波动太大，需要签名后广播。	1000
ExchangeWithdraw	对交易对撤资，需要签名后广播。	1000
ExchangeTransaction	参与交易对交易，需要签名后广播。	1000
GetAssetIssueByAccount	查询账户发行的TRC10通证。	1000
GetAccountNet	查询账户带宽信息。	1000
GetAccountResource	查询账户的资源信息（带宽、能量）。	1000

API方法	说明	流控值（次/s）
GetAssetIssueByName	根据通证名称查询TRC10通证。	200
GetAssetIssueListByName	根据名称返回同名的所有TRC 10代币列表。	200
GetAssetIssueById	根据ID查询TRC10通证。	1000
GetNowBlock	查询最新块。（请使用GetNowBlock2）	1000
GetNowBlock2	查询最新块。	1000
GetBlockByNum	通过高度查询区块内容。（请使用GetBlockByNum2）	15
GetBlockByNum2	通过高度查询区块内容。	15
GetTransactionCountByBlockNum	获取指定块中的交易计数。	1000
GetBlockById	通过区块ID（即区块哈希）查询区块。	15
GetBlockByLimitNext	查询指定范围的区块。（请使用GetBlockByLimitNext2）	10
GetBlockByLimitNext2	查询指定范围的区块。	10
GetBlockByLatestNum	查询最新的若干个区块。（请使用GetBlockByLatestNum2）	10
GetBlockByLatestNum2	查询最新的若干个区块。	10
GetTransactionById	按交易哈希查询交易。	1000
DeployContract	部署合约，返回 Transaction Extention，其中包含未签名的交易。	1000
GetContract	查询链上的合约信息，包括合约的bytecode、ABI、配置参数等。	300
GetContractInfo	查询链上的合约信息。与wallet/getcontract接口不同，该接口不仅返回bytecode还会返回合约的runtime bytecode。runtime bytecode相比bytecode，不包含构造函数以及构造函数的参数信息。	200

API方法	说明	流控值（次/s）
TriggerContract	调用智能合约，返回 TransactionExtention，需要签名后广播。	1000
TriggerConstantContract	调用合约只读函数，也可以调用合约非只读函数，用于预判交易是否可以执行成功或者预估交易的能量消耗，也可以预估合约部署消耗的能量。	1000
EstimateEnergy	预估智能合约调用交易或部署交易执行成功需要提供的能量。	1000
ClearContractAbi	将合约的 ABI 设置为空。返回未签名交易，需要签名后广播。	1000
ListWitnesses	返回所有超级代表的列表。	250
GetDelegatedResource	查看在stake1.0阶段一个账户代理给另外一个账户的资源情况。（请使用GetDelegatedResourceV2）	1000
GetDelegatedResourceV2	查看在stake1.0阶段一个账户代理给另外一个账户的资源情况。	1000
GetDelegatedResourceAccountIndex	查看在stake1.0阶段一个账户给哪些账户代理了资源，以及哪些账户为该账户代理了资源。	1000
GetDelegatedResourceAccountIndexV2	查询在Stake2.0阶段，某地址的资源委托索引。	1000
GetCanDelegatedMaxSize	查询目标地址中指定类型资源的可代理数量，单位为sun。	1000
GetAvailableUnfreezeCount	查询Stake2.0机制下，当前解质押剩余次数。	1000
GetCanWithdrawUnfreezeAmount	查询在某时间点可以提取的解质押本金数量。	1000
ListProposals	查询所有提案并返回提案信息。	300
GetProposalById	根据ID查询提案并返回提案详细信息。	1000
ListExchanges	查询所有交易对。	400
GetExchangeById	根据id查询交易对。	1000
GetChainParameters	查询当前所有提案参数，超级代表可以提议修改这些参数。	1000
GetAssetIssueList	查询所有TRC10通证列表。	5

API方法	说明	流控值（次/s）
GetPaginatedAssetIssueList	分页查询TRC10通证列表。	20
GetNextMaintenanceTime	返回下个计票时间点的时间戳（毫秒）。	1000
GetTransactionInfoById	查询交易的 Info 信息，包括交易的 fee 信息、所在区块和虚拟机log等。	1000
AccountPermissionUpdate	修改账户权限。	1000
GetTransactionSignWeight	查询交易签权重。	1000
GetTransactionApprovedList	根据交易内容和签名信息计算得到为交易签名的账户地址列表，可用交易验签。	1000
GetNodeInfo	查询当前节点的信息。	700
GetRewardInfo	查询用户未被提取的投票奖励。	1000
GetBrokerageInfo	查询超级代表佣金比例。	1000
UpdateBrokerage	更新 SR 佣金比例，需要签名后广播。	1000
GetTransactionInfoByBlockNum	获取特定区块的所有交易 Info 信息。	150
GetBurnTrx	查询自从第54号委员会提议生效后，因链上交易手续费而销毁的TRX数量。	1000
GetTransactionFromPending	从pending pool中获取交易详细信息。	1000
GetTransactionListFromPending	获取pending pool中交易列表信息。	1000
GetPendingSize	获取pending pool队列的大小。	1000
GetBlock	根据区块高度或者区块哈希查询区块头信息或者整个区块信息。	50
UnDelegateResource	在Stake 2.0机制下，取消为目标地址代理的带宽或者能量。	1000

API方法	说明	流控值（次/s）
GetBandwidthPrices	查询历史带宽单价。	1000
GetEnergyPrices	查询历史能量单价。	1000
GetMemoFee	获取交易备注费用。	1000

表 2-8 可用波场 API 列表-gRPC solidity

API方法	说明	流控值（次/s）
GetAccount	查询一个账号的信息，包括TRX余额、TRC10余额、质押以获取资源情况、投票情况以及权限等。	600
ListWitnesses	返回所有超级代表的列表。	250
GetAssetIssueList	查询所有TRC10通证列表。	5
GetPaginatedAssetIssueList	分页查询TRC10通证列表。	20
GetAssetIssueByName	根据通证名称查询TRC10通证。	200
GetAssetIssueListByName	根据名称返回同名的所有TRC 10代币列表。	200
GetAssetIssueById	根据ID查询TRC10通证。	1000
GetNowBlock	查询最新块。（请使用GetNowBlock2）	1000
GetNowBlock2	查询最新块。	1000
GetBlockByNum	通过高度查询区块内容。（请使用GetBlockByNum2）	15
GetBlockByNum2	通过高度查询区块内容。	15

API方法	说明	流控值（次/s）
GetTransactionCountByBlockNum	获取指定块中的交易计数。	1000
GetDelegatedResource	查看在stake1.0阶段一个账户代理给另外一个账户的资源情况。（请使用GetDelegatedResourceV2）	1000
GetDelegatedResourceV2	查看在stake1.0阶段一个账户代理给另外一个账户的资源情况。	1000
GetDelegatedResourceAccountIndex	查看在stake1.0阶段一个账户给哪些账户代理了资源，以及哪些账户为该账户代理了资源。	1000
GetDelegatedResourceAccountIndexV2	查询在Stake2.0阶段，某地址的资源委托索引。	1000
GetCanDelegatedMaxSize	查询目标地址中指定类型资源的可代理数量，单位为sun。	1000
GetAvailableUnfreezeCount	查询Stake2.0机制下，当前解质押剩余次数。	1000
GetCanWithdrawUnfreezeAmount	查询在某时间点可以提取的解质押本金数量。	1000
GetExchangeById	根据id查询交易对。	1000
ListExchanges	查询所有交易对。	400
GetTransactionById	按交易哈希查询交易。	1000
GetTransactionInfoById	查询交易的 Info 信息，包括交易的 fee 信息、所在区块和虚拟机log等。	1000
GetRewardInfo	查询用户未被提取的投票奖励。	1000
GetBrokerageInfo	查询超级代表佣金比例。	1000
TriggerConstantContract	调用合约只读函数，也可以调用合约非只读函数，用于预判交易是否可以执行成功或者预估交易的能量消耗，也可以预估合约部署消耗的能量。	1000
EstimateEnergy	预估智能合约调用交易或部署交易执行成功需要提供的能量。	1000

API方法	说明	流控值（次/s）
GetTransactionInfoByBlockNum	获取特定区块的所有交易 Info 信息。	150
GetBurnTrx	查询自从第54号委员会提议生效后，因链上交易手续费而销毁的TRX数量。	1000
GetBlock	根据区块高度或者区块哈希查询区块头信息或者整个区块信息。	50
GetBandwidthPrices	查询历史带宽单价。	1000
GetEnergyPrices	查询历史能量单价。	1000

2.6.2 共享版

表 2-9 可用波场 API 列表-wallet

API方法	类型	说明	计算单元(CU)
/wallet/validateaddress	POST	检查地址是否格式正确。	5
/wallet/broadcasttransaction	POST	广播签名后的交易。	5
/wallet/broadcasthex	POST	直接广播 protobuf 序列化后的十六进制 Transaction。	4
/wallet/createtransaction	POST	创建交易。	9
/wallet/createaccount	POST	激活一个链上账户。	10
/wallet/getaccount	POST	查询一个账号的信息，包括TRX余额、TRC10余额、质押以获取资源情况、投票情况以及权限等。	9
/wallet/updateaccount	POST	修改账户的名称。	9

API方法	类型	说明	计算单元(CU)
/wallet/accountpermissionupdate	POST	修改账户权限。	12
/wallet/getaccountresource	POST	查询账户的资源信息（带宽、能量）。	7
/wallet/getaccountnet	POST	查询账户带宽信息。	7
/wallet/unfreezebalance	POST	解锁在Stake1.0期间质押的TRX，释放所得到的带宽或能量以及TP，同时会自动取消所有投票。	5
/wallet/getdelegatedresource	POST	查看在stake1.0阶段一个账户代理给另外一个账户的资源情况。	5
/wallet/getdelegatedresourceaccountindex	POST	查看在stake1.0阶段一个账户给哪些账户代理了资源，以及哪些账户为该账户代理了资源。	40
/wallet/freezebalancev2	POST	在Stake2.0质押机制下，质押TRX以获取带宽或者能量，同时根据质押额度获得等值投票权(TP)。	10
/wallet/unfreezebalancev2	POST	解锁通过Stake2.0机制质押的TRX，释放所相应数量的带宽和能量，同时回收相应数量的投票权(TP)。	7
/wallet/delegateresource	POST	在Stake 2.0 机制下，将带宽或者能量资源代理给其它账户。	5
/wallet/undelegateresource	POST	在Stake 2.0机制下，取消为目标地址代理的带宽或者能量。	7
/wallet/withdrawexpireunfreeze	POST	提取已过锁定期的解质押的本金。	7
/wallet/getavailableunfreezecount	POST	查询Stake2.0机制下，当前解质押剩余次数。	5
/wallet/getcanwithdrawunfreezeamount	POST	查询在某时间点可以提取的解质押本金数量。	6
/wallet/getcandelegatedmaxsize	POST	查询目标地址中指定类型资源的可代理数量，单位为sun。	7
/wallet/getdelegatedresourcev2	POST	查询在Stake2.0机制下，某地址代理给目标地址的资源情况。	5

API方法	类型	说明	计算单元(CU)
/wallet/getdelegatedresourceaccountindexv2	POST	查询在Stake2.0阶段，某地址的资源委托索引账号。	40
/wallet/getblock	POST	根据区块高度或者区块哈希查询区块头信息或者整个区块信息。	392
/wallet/getblockbynum	POST	通过高度查询区块内容。	192
/wallet/getblockbyid	POST	通过区块ID（即区块哈希）查询区块。	392
/wallet/getblockbylatestnum	POST	查询最新的若干个区块。	941
/wallet/getblockbylimitnext	POST	查询指定范围的区块。	392
/wallet/getnowblock	GET	查询最新块。	542
/wallet/gettransactionbyid	POST	按交易哈希查询交易。	52
/wallet/gettransactioninfobyid	POST	查询交易的 Info 信息，包括交易的 fee 信息、所在区块和虚拟机log等。	19
/wallet/gettransactioninfobyblocknum	POST	获取特定区块的所有交易 Info 信息。	139
/wallet/getchainparameters	GET	查询当前所有提案参数，超级代表可以提议修改这些参数。	13
/wallet/getenergyprices	GET	查询历史能量单价。	8
/wallet/getbandwidthprices	GET	查询历史带宽单价。	6
/wallet/getburntrx	GET	查询自从第54号委员会提议生效后，因链上交易手续费而销毁的TRX数量。	6
/wallet/getapprovedlist	POST	根据交易内容和签名信息计算得到为交易签名的账户地址列表，可用交易验签。	20
/wallet/getassetissuebyaccount	POST	查询账户发行的TRC10通证。	523
/wallet/getassetissuebyid	POST	根据ID查询TRC10通证。	9
/wallet/getassetissuebyname	POST	根据通证名称查询TRC10通证。	570

API方法	类型	说明	计算单元(CU)
/wallet/getassetissuelist	GET	查询所有TRC10通证列表。	4706
/wallet/getassetissuelistbyname	POST	根据名称返回同名的所有TRC 10代币列表。	509
/wallet/getpaginatedassetissuelist	POST	分页查询TRC10通证列表。	784
/wallet/transferasset	POST	转账TRC10通证。	20
/wallet/createassetissue	POST	发行TRC10通证。	20
/wallet/participateassetissue	POST	参与TRC10通证发行。	20
/wallet/unfreezeasset	POST	解锁已经结束质押期的 TRC10通证。	20
/wallet/updateasset	POST	修改TRC10通证基本信息。	20
/wallet/getcontract	POST	查询链上的合约信息，包括合约的bytecode、 ABI、配置参数等。	84
/wallet/getcontractinfo	POST	查询链上的合约信息。与wallet/getcontract接口不同，该接口不仅返回bytecode还会返回合约的runtime bytecode。runtime bytecode相比bytecode，不包含构造函数以及构造函数的参数信息。	114
/wallet/triggersmartcontract	POST	调用智能合约，返回TransactionExtention，需要签名后广播。	20
/wallet/triggerconstantcontract	POST	调用合约只读函数，也可以调用合约非只读函数，用于预判交易是否可以执行成功或者预估交易的能量消耗，也可以预估合约部署消耗的能量。	20
/wallet/deploycontract	POST	部署合约，返回 TransactionExtention，其中包含未签名的交易。	14
/wallet/updatesetting	POST	更新合约的consume_user_resource_percent 配置，返回未签名交易，需要签名后广播。	20

API方法	类型	说明	计算单元(CU)
/wallet/updateenergylimit	POST	更新合约的 origin_energy_limit，返回未签名交易，需要签名后广播。	20
/wallet/clearabi	POST	将合约的 ABI 设置为空。返回未签名交易，需要签名后广播。	20
/wallet/estimateenergy	POST	预估智能合约调用交易或部署交易执行成功需要提供的能量。	13
/wallet/createwitness	POST	申请成为超级代表， 返回申请超级代表的Transaction，需要签名后广播。	20
/wallet/updatewitness	POST	修改witness配置信息中的 URL，需要签名后广播。	20
/wallet/listwitnesses	GET	返回所有超级代表的列表。	20
/wallet/votewitnessaccount	POST	对超级代表进行投票，返回投票的Transaction， 需要签名后广播。	20
/wallet/updateBrokerage	POST	更新 SR 佣金比例，需要签名后广播。	7
/wallet/getBrokerage	POST	查询超级代表佣金比例。	5
/wallet/getReward	POST	查询用户未被提取的投票奖励。	5
/wallet/withdrawbalance	POST	超级代表或用户提取奖励，每 24 小时可调用一次。超级代表将 allowance 中的余额提取到账户中，用户将投票奖励提取到自己的账户中。	20
/wallet/getnextmaintenancetime	GET	返回下个计票时间点的时间戳（毫秒）。	20
/wallet/proposalcreate	POST	创建提案交易，需要签名后广播。	5
/wallet/proposalapprove	POST	批准提案，需要签名后广播。	20
/wallet/proposaldelete	POST	删除提案，需要签名后广播。	20
/wallet/listproposals	GET	查询所有提案并返回提案信息。	116
/wallet/getproposalbyid	POST	根据ID查询提案并返回提案详细信息。	10
/wallet/exchangecreate	POST	创建交易对，需要签名后广播。警告：成功执行，签署和广播此 API 调用将从用户的帐户中扣除 1024 TRX。	20

API方法	类型	说明	计算单元(CU)
/wallet/exchangeinject	POST	给交易对注资，注资后可以防止交易对价格波动太大，需要签名后广播。	20
/wallet/exchangewithdraw	POST	对交易对撤资，需要签名后广播。	20
/wallet/exchangetransaction	POST	参与交易对交易，需要签名后广播。	20
/wallet/getexchangebyid	POST	根据id查询交易对。	8
/wallet/listexchanges	GET	查询所有交易对。	79
/wallet/gettransactionlistfrompending	GET	获取pending pool中交易列表信息。	20
/wallet/gettransactionfrompending	POST	从pending pool中获取交易详细信息。	9
/wallet/getpendingsize	GET	获取pending pool队列的大小。	7
/wallet/getsignweight	POST	查询交易签名权重。	20

表 2-10 可用波场 API 列表-walletsolidity

API方法	类型	说明	计算单元(CU)
/walletsolidity/gettransactionbyid	POST	按交易哈希查询交易（已固化状态）。	28
/walletsolidity/gettransactioninfobyid	POST	查询交易的 Info 信息，包括交易的 fee 信息、 所在区块、虚拟机log等（已固化状态）。	10
/walletsolidity/gettransactioninfobyblocknum	POST	获取特定区块的所有交易 Info 信息（已固化状态）。	18
/walletsolidity/getblock	POST	根据区块高度或者区块哈希查询区块头信息或者整个区块信息（已固化状态）。	392
/walletsolidity/getblockbyid	POST	通过区块ID（即区块哈希）查询区块（已固化状态）。	376
/walletsolidity/getblockbylatestnum	POST	查询最新的若干个区块（已固化状态）。	965

API方法	类型	说明	计算单元(CU)
/walletsolidity/getblockbylimitnext	POST	查询指定范围的区块（已固化状态）。	376
/walletsolidity/getblockbynum	POST	查询确认指定块是否被固化。	192
/walletsolidity/getnowblock	GET	查询当前最新区块（已固化状态）。	542
/walletsolidity/getaccount	POST	获取账户信息（已固化状态）。	9
/walletsolidity/getdelegatedresourcev2	POST	Stake 2.0 API：查询某地址代理给目标地址的资源情况（已固化状态）。	5
/walletsolidity/getavailableunfreezecount	POST	Stake 2.0 API: 查询当下解质押剩余次数（已固化状态）。	5
/walletsolidity/getcanwithdrawunfreezeamount	POST	Stake 2.0 API: 查询在某时间点可以提取的解质押本金数量（已固化状态）。	6
/walletsolidity/getdelegatedresourceaccountindexv2	POST	Stake 2.0 API：查询某地址的资源委托索引（已固化状态）。返回两个列表，一个是该帐户将资源委托给的地址列表（toAddress）；另一个是将资源委托给该帐户的地址列表（fromAddress）。	7
/walletsolidity/getburntrx	GET	查询自从第54号委员会提议生效后，因链上交易手续费而销毁的TRX数量（已固化状态）。	5
/walletsolidity/triggerconstantcontract	POST	既可以调用合约只读函数（view 或 pure修饰的函数），用于查询合约已固化状态数据，也可以调用合约非只读函数，用于在已固化状态下预判交易是否可以执行成功或者预估交易的能量消耗。	20
/walletsolidity/estimateenergy	POST	在已固化状态下，预估智能合约交易执行成功需要提供的能量。	13
/walletsolidity/getassetissuebyid	POST	根据ID查询TRC10通证（已固化状态）。	34
/walletsolidity/getassetissuebyname	POST	根据通证名称查询TRC10通证（已固化状态）。	36
/walletsolidity/getassetissuelist	GET	查询所有TRC10通证列表（已固化状态）。	4183

API方法	类型	说明	计算单元(CU)
/walletsolidity/getassetissuelistbyname	POST	根据名称返回同名的所有TRC10代币列表（已固化状态）。	495
/walletsolidity/getpaginatedassetissuelist	POST	分页查询TRC10通证列表（已固化状态）。	784
/walletsolidity/listwitnesses	GET	返回所有超级代表的列表（已固化状态）。	111
/walletsolidity/getexchangebyid	POST	根据id查询交易对（已固化状态）。	6
/walletsolidity/listexchanges	GET	查询所有交易对（已固化状态）。	79
/walletsolidity/getenergyprices	GET	查询历史能量单价。	17
/walletsolidity/getcandelegatedmaxsize	POST	Stake 2.0 API: 查询目标地址中指定类型资源的可代理数量（已固化状态），单位为sun。	7
/walletsolidity/gettransactioncountbyblocknum	POST	按区块号查询区块内交易数量（已固化状态）。	18

表 2-11 可用波场 API 列表-jsonrpc

API方法	类型	说明	计算单元(CU)
eth_accounts	POST	返回客户端拥有的地址列表，tron将返回空列表。	13
eth_blockNumber	POST	获取最新区块号。	10
eth_call	POST	立即执行消息调用，而不在区块链上创建交易，即triggerConstantContract。	20
eth_chainId	POST	返回TRON chainId，TRON chainId为创世块哈希的最后四个字节。	1
eth_estimateGas	POST	通过triggerConstantContract预估能量消耗。	1000
eth_gasPrice	POST	获取当前的能量单价（以sun为单位）。	13

API方法	类型	说明	计算单元(CU)
eth_getBalance	POST	获取给定地址的账户余额。	15
eth_getBlockByHash	POST	根据区块哈希获取区块信息。	45
eth_getBlockByNumber	POST	根据区块号获取区块信息。	20
eth_getBlockTransactionCountByHash	POST	根据区块哈希获取区块内的交易数量。	13
eth_getBlockTransactionCountByNumber	POST	根据区块号获取区块内的交易数量。	12
eth_getCode	POST	获取给定智能合约的runtime code。	40
eth_getStorageAt	POST	返回某地址的指定位置存储的内容，可用于获取某个合约中某个变量的值。	15
eth_getTransactionByBlockHashAndIndex	POST	根据区块哈希，获取区块的第index个交易。	17
eth_getTransactionByBlockNumberAndIndex	POST	根据区块号，获取区块的第index个交易。	18
eth_getTransactionByHash	POST	根据交易哈希获取交易信息。	40
eth_getTransactionReceipt	POST	查询交易的 Info 信息，包括交易的 fee 信息、所在区块和虚拟机log等。	19
eth_syncing	POST	获取节点的同步状态。	1
eth_getLogs	POST	返回与给定过滤条件匹配的所有事件。	75
net_listening	POST	查询客户端是否处于监听网络连接的状态。	1
net_version	POST	返回创世块的哈希值。	1
web3_clientVersion	POST	返回当前节点的版本。	1
web3_sha3	POST	计算给定数据的Keccak-256值（不是标准的SHA3-256）。	5
buildTransaction	POST	创建transaction，交易类型不同，参数不同。	13

3 Polygon PoS

3.1 Polygon PoS 介绍

Polygon PoS是一种“Layer-2”或“侧链”以太坊扩展解决方案，与以太坊主网并行运行。Polygon PoS支持使用最广泛的以太坊扩展生态系统，该生态系统提供 EVM 兼容性和终极用户体验，如今以接近零的 Gas 费用进行快速交易。

Polygon官方链接：开发者中心，白皮书

用户可以通过使用华为云公链节点引擎，来提升区块链使用与开发的效率，增强其稳定性与私密性。**华为云将永远不会收集用户的区块链地址。**

说明

- **支持网络**
  - Polygon PoS：HTTP、WebSocket
- **Polygon PoS API列表**

3.2 HTTP 请求示例

3.2.1 使用 cURL 发送 HTTP API 请求

Request example:

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"eth_getBlockByNumber","params":["0xc5043f",false],"id":1,"jsonrpc":"2.0"}'
```

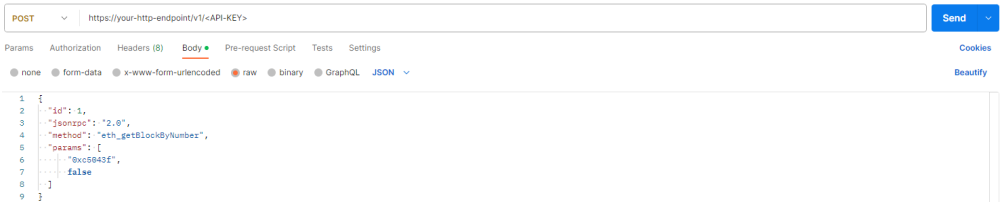
Response example:

```
{
    "jsonrpc": "2.0",
    "id": 1,
    "result": {
        "number": "0xc5043f",
        "hash": "0x6c2a069ee47f1fcb83a64c8f3193944199545e91928097d4bc79265f62135040",
        "transactions": [
            "0xd3cee91cc3f05d9228bf2ae99629af0bef22ced9c00fabb7d90be3b1dd029f0b",
```

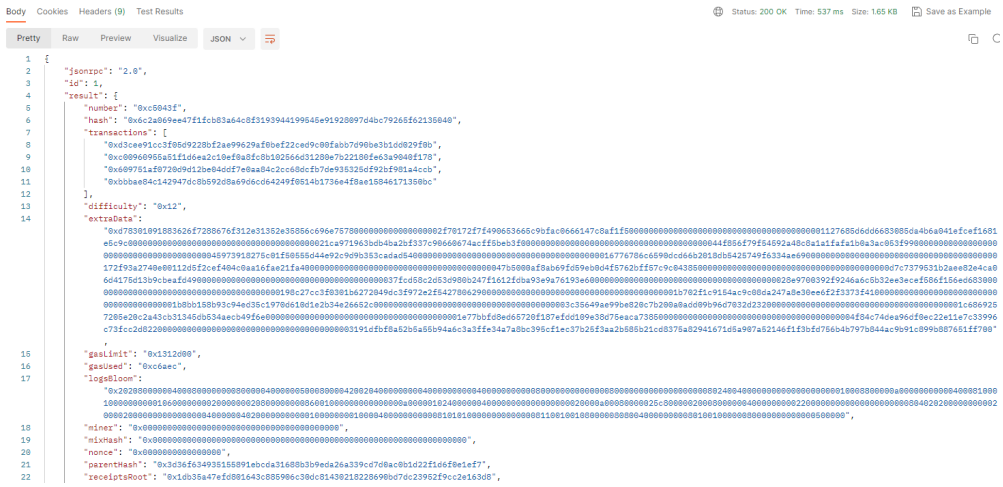
```
            "0xc00960955a51f1d6ea2c10ef0a8fc8b102566d31280e7b22180fe63a9040f178",
            "0x609751af0720d9d12be04ddf7e0aa84c2cc68dcfb7de935325df92bf981a4ccb",
            "0xbbbae84c142947dc8b592d8a69d6cd64249f0514b1736e4f8ae15846171350bc"
        ],
        "difficulty": "0x12",
        "extraData":
"0xd78301091883626f7288676f312e31352e35856c696e757800000000000000000002f70172f7f490
653665c9bfac0666147c8af1f500000000000000000000000000000000000000000001127685d6dd6683
085da4b6a041efcef1681e5c9c000000000000000000000000000000000000000021ca971963bdb4b
a2bf337c90660674acff5beb3f000000000000000000000000000000000000000044f856f79f54592a
48c8a1a1fafa1b0a3ac053f990000000000000000000000000000000000000045973918275c01f5
0555d44e92c9d9b353cadad5400000000000000000000000000000000000016776786c6590dc
d66b2018db5425749f6334ae690000000000000000000000000000000000000172f93a2740e00
112d5f2cef404c0aa16fae21fa400000000000000000000000000000000000047b5000af8ab69f
d59eb0d4f5762bff57c9c043850000000000000000000000000000000000000d7c7379531b2aee
82e4ca06d4175d13b9cbeafd490000000000000000000000000000000000037fcd58c2d53d98
0b247f1612fdba93e9a76193e600000000000000000000000000000000000028e9700392f9246
a6c5b32ee3ecef586f156ed68300000000000000000000000000000000000198c27cc3f0301b6
272049dc3f972e2f5427806290000000000000000000000000000000001b702f1c9154ac9c
08da247a8e30ee6f2f3373f41000000000000000000000000000000001b8bb158b93c94ed
35c1970d610d1e2b34e26652c0000000000000000000000000000000003c35649ae99be82
0c7b200a0add09b96d7032d232000000000000000000000000000000001c6869257205e2
0c2a43cb31345db534aecb49f6e0000000000000000000000000000001e77bbfd8ed657
20f187efdd109e38d75eaca738500000000000000000000000000000004f84c74dea96df
0ec22e11e7c33996c73fcc2d822000000000000000000000000000003191dfbf8a52b5a
55b94a6c3a3ffe34a7a8bc395cf1ec37b25f3aa2b585b21cd8375a82941671d5a907a52146f1f3bfd7
56b4b797b844ac9b91c899b887651ff700",
        "gasLimit": "0x1312d00",
        "gasUsed": "0xc6aec",
        "logsBloom":
"0x2020800000400080000000800004000000500080000420020400000000400000000004000
0000000080000000000000800000000000000000802400400000000000000000001000880
0000a000000000040008100010000000001060000000020000002080000000860010000000
00000000a000001024000000400000000000000000020000a00080000025c80000200080000004
0000000002200000000000000000000804020200000000020000200000000000000004000000
4020000000000010000000010000400000000000810101000000000000081100100108000008
8004000000008010010000008000000000000500000",
        "miner": "0x0000000000000000000000000000000000000000",
        "mixHash":
"0x0000000000000000000000000000000000000000000000000000000000000000",
        "nonce": "0x0000000000000000",
        "parentHash":
"0x3d36f634935155891ebcda31688b3b9eda26a339cd7d0ac0b1d22f1d6f0e1ef7",
        "receiptsRoot":
"0x1db35a47efd801643c885906c30dc81430218228690bd7dc23952f9cc2e163d8",
        "sha3Uncles":
"0x1dcc4de8dec75d7aab85b567b6ccd41ad312451b948a7413f0a142fd40d49347",
        "size": "0xdb8",
        "stateRoot":
"0xfe50dacb41e2de9119bca1ee4f2345c8dc2da79b9171b6f61d858fba40f4cabe",
        "timestamp": "0x606b7a3a",
        "totalDifficulty": "0x6e6d156",
        "transactionsRoot":
"0x89889dccc152914e70a3675868d7784a1a9dbdfb1be7f6943f873ca9d07942a7",
        "uncles": []
    }
}
```


3.2.2 使用 post-man 发送 HTTP API 请求

Request example:



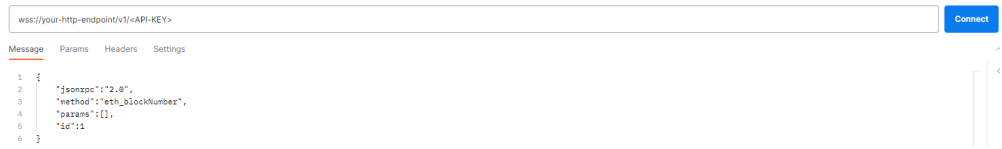
Response example:



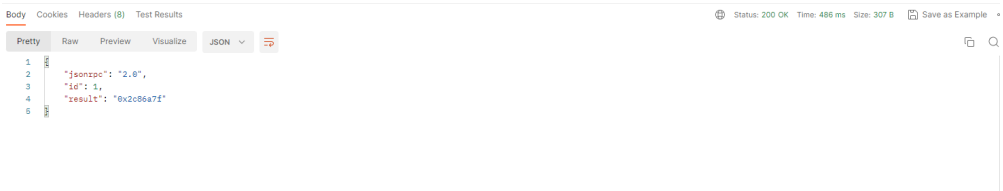
3.3 WebSocket 请求示例

3.3.1 使用 post-man 发送 JSON-RPC API 请求

Request example:



Response example:



3.4 Polygon PoS API 列表

3.4.1 专享版

表 3-1 可用 Polygon API 列表

API方法	说明	流控值（次/s）	
		8U32G	16U64G
eth_blockNumber	返回区块链的最新区块号。	30000	60000
eth_getBlockByNumber	返回与给定的区块号匹配的区块信息。	4000	35000
eth_getUncleByBlockNumberAndIndex	返回给定区块号和索引位置的叔区块信息。	30000	51000
eth_getUncleByBlockHashAndIndex	返回给定区块号和索引位置的叔区块信息。	30000	60000
eth_getUncleCountByBlockNumber	返回与给定区块编号匹配的区块中叔区块的数量。	30000	53000
eth_getUncleCountByBlockHash	返回与给定区块哈希匹配的区块中叔区块的数量。	30000	58000
eth_getBlockByHash	返回与给定区块哈希匹配的区块的信息。	9000	15000
eth_getTransactionByHash	根据交易哈希返回有关交易的信息。	20000	17000
eth_getTransactionCount	返回从某一地址发送的交易数。	25000	46000
eth_getTransactionByBlockHashAndIndex	返回给定交易哈希和交易索引位置的交易信息。	25000	43000
eth_getTransactionByBlockNumberAndIndex	返回给定区块号和交易索引位置的交易信息。	20000	41000
eth_getBlockTransactionCountByNumber	返回与给定区块编号匹配的区块的交易数。	25000	57000
eth_getBlockTransactionCountByHash	返回与给定块哈希匹配的区块的交易数。	25000	54000
eth_getTransactionReceipt	通过交易哈希返回交易的收据。	6000	17000
eth_getTransactionReceiptsByBlock	返回给定区块号或哈希的所有交易收据。	200	650

API方法	说明	流控值（次/s）	
		8U32G	16U64G
eth_sendRawTransaction	创建新的消息调用交易或为签名交易创建合约。	1000	2400
eth_call	立即执行新的消息调用，而不在区块链上创建交易。该API所消耗的计算单元为20。	15000	37000
eth_createAccessList	基于给定的交易对象创建 EIP2930 类型 accessList。返回交易读取和写入的地址以及存储密钥列表，但是发送者帐户和预编译除外。	1000	2400
eth_estimateGas	返回给定交易的所消耗的Gas的估计值。	50	720
eth_feeHistory	返回历史消耗的Gas信息的集合。	25000	42000
eth_maxPriorityFeePerGas	返回每个Gas的费用，这是您可以支付多少优先费用或“小费”的估计，以获得当前区块中包含的交易。	30000	54000
eth_gasPrice	返回当前的Gas价格（以 wei 为单位）。	30000	53000
eth_getBalance	返回给定地址的帐户余额。	25000	48000
eth_getStorageAt	返回给定地址的存储位置的值。	25000	47000
eth_accounts	返回客户端拥有的地址列表。	30000	53000
eth_getCode	返回给定地址处智能合约的已编译字节代码（如果有）。	8000	15000
eth_getProof	返回指定账户的账户和存储值，包括 Merkle 证明。	10000	17000
eth_getLogs	返回与给定过滤器对象匹配的所有日志的数组。	10000	14000

API方法	说明	流控值（次/s）	
		8U32G	16U64G
eth_getFilterChanges	过滤器的轮询方法，返回自上次轮询以来发生的日志数组。过滤器必须通过调用eth_newFilter、eth_newBlockFilter、eth_newPendingTransactionFilter来创建。	30000	58000
eth_getFilterLogs	返回与给定过滤器 ID 匹配的所有日志的数组。	500	1300
eth_newBlockFilter	在节点中创建一个过滤器，以在新区块到达时发出通知。	25000	30000
eth_newFilter	根据给定的过滤器选项创建过滤器对象，以在状态更改（日志）时发出通知。	25000	41000
eth_newPendingTransactionFilter	在节点中创建一个过滤器，以在新的待处理事务到达 Polygon 时发出通知。	30000	30000
eth_uninstallFilter	卸载具有给定 id 的过滤器。当不再需要观察时调用。此外，如果一段时间内没有通过eth_getFilterChanges请求过滤器，过滤器就会超时。	25000	55000
eth_getRootHash	返回指定区块范围的根哈希。	7000	20000
eth_syncing	返回当前的同步状态。	30000	62000
eth_chainId	返回当前配置的链 ID。	30000	58000
bor_getAuthor	返回指定块的作者。	20000	55000
bor_getCurrentProposer	返回当前提议者的地址。	30000	53000
bor_getCurrentValidators	返回当前验证者列表。	15000	38000
bor_getRootHash	返回指定区块范围的根哈希。	30000	53000

API方法	说明	流控值（次/s）	
		8U32G	16U64G
bor_getSignersAtHash	返回与指定块哈希匹配的块的所有签名者。	20000	44000
debug_getBadBlocks	返回客户端在网络上看到的最后一个坏块的列表。	30	130
debug_storageRangeAt	返回指定范围的合约存储。	25000	48000
debug_traceBlock	追踪指定的交易。	50	140
debug_traceBlockByHash	追踪指定区块哈希中的所有交易的详细调用信息，包括调用类型、发送者、接收者地址、交易值、gas、输入数据、输出数据等。	50	140
debug_traceBlockByNumber	追踪指定区块号中的所有交易的详细调用信息，包括调用类型、发送者、接收者地址、交易值、gas、输入数据、输出数据等。	50	120
debug_traceCall	通过在给定块执行的上下文中执行eth调用来返回可能的跟踪结果。	15000	31000
debug_traceTransaction	追踪指定的交易。	25	250
web3_sha3	返回给定数据的Keccak-256 编码结果（不是标准化SHA3-256）。	250000	53000
web3_clientVersion	返回当前客户端的版本。	25000	56000
txpool_content	返回所有挂起和排队的交易。	5	10
txpool_inspect	返回所有挂起和排队的交易的文本摘要。	15	30
txpool_status	返回处于待处理（Pending）状态和排队（Queued）状态的事务数。	3000	6500
net_version	返回当前网络id。	30000	58000

API方法	说明	流控值（次/s）	
		8U32G	16U64G
net_listening	当客户端正在主动侦听网络连接时为true。	30000	51000
eth_subscribe	为特定事件创建新订阅。节点返回订阅 ID。对于与订阅匹配的每个事件，将发送包含相关数据的通知以及订阅ID。	1000	1000
eth_unsubscribe	通过使用订阅 ID 调用此方法来取消订阅。它返回一个布尔值，指示订阅已成功取消。	1000	1000
nes_sendGasOptimizedTransaction	返回用于查询该Gas优化交易状态的id。	100	250
nes_getGasOptimizedTransactionStatus	返回增强交易的状态。	1500	1500

3.4.2 共享版

3.4.2.1 Ethereum JSON-RPC API

3.4.2.1.1 eth_blocknumber

简介

返回区块链的最新区块号。该API所消耗的计算单元为10。

参数说明

此方法不接受任何参数。

返回值

十六进制编码的最新区块号。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"eth_blockNumber","params":[],"id":1,"jsonrpc":"2.0"}'
```

3.4.2.1.2 eth_getBlockByNumber

简介

返回与给定的区块号匹配的区块信息。该API所消耗的计算单元为20。

参数说明

参数	类型	说明
区块编号	String	十六进制的区块编号，或者是字符串 "earliest"、"latest"、"pending"。
交易详细信息标志	Bool	当此值为 true 时，该方法返回完整的交易对象，否则，它仅返回交易的哈希。

返回值

- Object - 区块对象，如果未找到区块，则为 null。区块对象包含以下字段：
  - number: 编码为十六进制的请求块的块号。如果是一个处于待处理（ Pending ）状态的区块，则为空。
  - hash: 区块的哈希。如果是一个处于待处理（ Pending ）状态的区块，则为空。
  - parentHash: 父区块的哈希。
  - nonce: 生成的工作量证明的哈希值。如果是一个处于待处理（ Pending ）状态的区块，则为空。
  - sha3Uncles: 区块中叔区块数据的 SHA3。
  - logsBloom: 区块日志的布隆过滤器。如果是一个处于待处理（ Pending ）状态的区块，则为空。
  - transactionsRoot: 区块中交易树的根。
  - stateRoot： 区块的最终状态树的根。
  - receiptsRoot： 区块的收据树的根。
  - miner： 获得采矿奖励的受益人的地址。
  - difficulty：此区块的难度。
  - totalDifficulty：直到这个区块时，链的总难度。
  - extraData：此区块的“额外数据”字段。
  - size：此区块的大小（以字节为单位）。
  - gasLimit：此区块中允许的最大gas。
  - gasUsed：此区块中所有交易的总使用gas。
  - timestamp：整理区块时的 unix 时间戳。
  - transactions：交易对象的数组，或 32 字节的交易哈希，具体取决于最后一个给定的参数。
  - uncles：叔区块哈希数组。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"eth_getBlockByNumber","params":["0xc5043f",false],"id":1,"jsonrpc":"2.0"}'
```

3.4.2.1.3 eth_getUncleByBlockNumberAndIndex

简介

按区块编号和叔区块索引位置返回有关叔区块的信息。该API所消耗的计算单元为14。

参数说明

参数	类型	说明
区块编号或标签	String	想要查询的区块的十六进制的编号，或者是字符串"earliest"、"latest"、"pending"。
叔区块的索引位置	String	叔区块的十六进制的区块编号。

返回值

- Object - 区块对象，如果未找到区块，则为 null。区块对象包含以下字段：
  - number: 编码为十六进制的请求块的块号。如果是一个处于待处理（Pending）状态的区块，则为空。
  - hash: 区块的哈希。如果是一个处于待处理（Pending）状态的区块，则为空。
  - parentHash: 父区块的哈希。
  - nonce: 生成的工作量证明的哈希值。如果是一个处于待处理（Pending）状态的区块，则为空。
  - sha3Uncles: 区块中叔区块数据的 SHA3。
  - logsBloom: 区块日志的布隆过滤器。如果是一个处于待处理（Pending）状态的区块，则为空。
  - transactionsRoot: 区块中交易树的根。
  - stateRoot： 区块的最终状态树的根。
  - receiptsRoot： 区块的收据树的根。
  - miner： 获得采矿奖励的受益人的地址。
  - difficulty：此区块的难度。
  - totalDifficulty： 直到这个区块时，链的总难度。
  - extraData：此区块的“额外数据”字段。
  - size：此区块的大小（以字节为单位）。
  - gasLimit：此区块中允许的最大gas。
  - gasUsed：此区块中所有交易的总使用gas。

- timestamp：整理区块时的 unix 时间戳。
- transactions：交易对象的数组，或 32 字节的交易哈希，具体取决于最后一个给定的参数。
- uncles：叔区块哈希数组。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
 -X POST \
 -H "Content-Type: application/json" \
 --data '{"method":"eth_getUncleByBlockNumberAndIndex","params":["latest","0x0"],"id":1,"jsonrpc":"2.0"}'
```

3.4.2.1.4 eth_getUncleByBlockHashAndIndex

简介

按区块哈希和叔区块索引位置返回有关叔区块的信息。该API所消耗的计算单元为12。

参数说明

参数	类型	说明
区块哈希	String	想要查询的区块的哈希值。
叔区块的索引位置	String	叔区块的十六进制的区块编号。

返回值

- Object - 区块对象，如果未找到区块，则为 null。区块对象包含以下字段：
  - number: 编码为十六进制的请求块的块号。如果是一个处于待处理（Pending）状态的区块，则为空。
  - hash: 区块的哈希。如果是一个处于待处理（Pending）状态的区块，则为空。
  - parentHash: 父区块的哈希。
  - nonce: 生成的工作量证明的哈希值。如果是一个处于待处理（Pending）状态的区块，则为空。
  - sha3Uncles: 区块中叔区块数据的 SHA3。
  - logsBloom: 区块日志的布隆过滤器。如果是一个处于待处理（Pending）状态的区块，则为空。
  - transactionsRoot: 区块中交易树的根。
  - stateRoot： 区块的最终状态树的根。
  - receiptsRoot： 区块的收据树的根。
  - miner： 获得采矿奖励的受益人的地址。
  - difficulty：此区块的难度。
  - totalDifficulty： 直到这个区块时，链的总难度。
  - extraData：此区块的“额外数据”字段。

- size：此区块的大小（以字节为单位）。
- gasLimit：此区块中允许的最大gas。
- gasUsed：此区块中所有交易的总使用gas。
- timestamp：整理区块时的 unix 时间戳。
- transactions：交易对象的数组，或 32 字节的交易哈希，具体取决于最后一个给定的参数。
- uncles：叔区块哈希数组。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"eth_getUncleByBlockHashAndIndex","params":
[0xc6ef2fc5426d6ad6fd9e2a26abeab0aa2411b7ab17f30a99d3cb96aed1d1055b",
"0x0"],"id":1,"jsonrpc":"2.0"}'
```

3.4.2.1.5 eth_getUncleCountByBlockNumber

简介

返回与给定区块编号匹配的区块中叔区块的数量。该API所消耗的计算单元为13。

参数说明

参数	类型	说明
区块编号	String	想要查询的区块的十六进制的编号。

返回值

区块中叔区块的数量，以十六进制为编码。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"eth_getUncleCountByBlockNumber","params":["0xc5043f"],"id":1,"jsonrpc":"2.0"}'
```

3.4.2.1.6 eth_getUncleCountByBlockHash

简介

返回与给定区块哈希匹配的区块中叔区块的数量。该API所消耗的计算单元为12。

参数说明

参数	类型	说明
区块哈希	String	想要查询的区块的哈希值。

返回值

区块中叔区块的数量，以十六进制为编码。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"eth_getUncleCountByBlockHash","params":
["0x81e807e7a6031d9f103eeee2a2edc5994c3432ee1e3227c66ff78eef30ea1dec"],"id":1,"jsonrpc":"2.0"}'
```

3.4.2.1.7 eth_getBlockByHash

简介

返回与给定区块哈希匹配的区块的信息。该API所消耗的计算单元为45。

参数说明

参数	类型	说明
区块哈希	String	想要查询的区块的哈希值。
事务详细信息标志	Bool	当此值为 true 时，该方法返回完整的交易对象，否则，它仅返回交易的哈希。

返回值

- Object - 区块对象，如果未找到区块，则为 null。区块对象包含以下字段：
  - number: 编码为十六进制的请求块的块号。如果是一个处于待处理（Pending）状态的区块，则为空。
  - hash: 区块的哈希。如果是一个处于待处理（Pending）状态的区块，则为空。
  - parentHash: 父区块的哈希。
  - nonce: 生成的工作量证明的哈希值。如果是一个处于待处理（Pending）状态的区块，则为空。
  - sha3Uncles: 区块中叔区块数据的 SHA3。
  - logsBloom: 区块日志的布隆过滤器。如果是一个处于待处理（Pending）状态的区块，则为空。
  - transactionsRoot: 区块中交易树的根。
  - stateRoot： 区块的最终状态树的根。
  - receiptsRoot： 区块的收据树的根。
  - miner： 获得采矿奖励的受益人的地址。
  - difficulty：此区块的难度。
  - totalDifficulty： 直到这个区块时，链的总难度。

- extraData：此区块的“额外数据”字段。
- size：此区块的大小（以字节为单位）。
- gasLimit：此区块中允许的最大gas。
- gasUsed：此区块中所有交易的总使用gas。
- timestamp：整理区块时的 unix 时间戳。
- transactions：交易对象的数组，或 32 字节的交易哈希，具体取决于最后一个给定的参数。
- uncles：叔区块哈希数组。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"eth_getBlockByHash","params":
["0x81e807e7a6031d9f103eeee2a2edc5994c3432ee1e3227c66ff78eef30ea1dec",false],"id":1,"jsonrpc":"2.0"}'
```

3.4.2.1.8 eth_getTransactionByHash

简介

根据交易哈希返回有关交易的信息。该API所消耗的计算单元为40。

参数说明

参数	类型	说明
交易哈希	String	要查询的交易的哈希值。

返回值

- Object - 交易对象，如果未找到交易，则为 null。交易对象包含以下字段：
  - blockHash：此交易所在的区块的哈希值。当它是待处理的（Pending）日志时为 null
  - blockNumber：此交易所在的区块号。当它是待处理的（Pending）日志时为 null
  - from：发件人的地址
  - gas：发送方提供的gas，编码为十六进制
  - gasPrice：发件人提供的 wei 格式的gas价格，编码为十六进制
  - maxFeePerGas：交易中设置的每种gas的最高值
  - maxPriorityFeePerGas：交易中设置的最高优先级gas
  - hash：交易的哈希值
  - input：与交易一起发送的数据
  - nonce：发送方在此交易之前进行的交易数，编码为十六进制
  - to：接收方的地址。当它是合约创建交易时为 null
  - transactionIndex：从中创建日志的交易索引位置的整数。当它是待处理的（Pending）日志时为 null

- value：以 wei 为单位的十六进制编码的转账数额
- type：交易类型
- accessList：交易计划访问的地址和存储密钥的列表
- chainId：交易的链 ID（如果有）
- v：签名的标准化 V 字段
- r：签名的 R 字段
- s：签名的 S 字段

请求样式

```
curl https://polygon-mainnet.shared-fullnode.bcs.ap-southeast-3.myhuaweicloud.com/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"eth_getTransactionByHash","params":
["0xb142342a7fd70602b7a0ba3688a41bfcbb4fbc3490c252ca48af2594619d220c"],"id":1,"jsonrpc":"2.0"}'
```

3.4.2.1.9 eth_getTransactionCount

简介

返回从某一地址发送的交易数。该API所消耗的计算单元为15。

参数说明

参数	类型	说明
地址	String	要检查的交易计数的地址。
区块编号	String	想要查询的区块的十六进制的编号，或者是字符串"earliest"、"latest"、"pending"。

返回值

从地址发送的十六进制编码的交易数量

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"eth_getTransactionCount","params":
["0x8D97689C9818892B700e27F316cc3E41e17fBeb9", "latest"],"id":1,"jsonrpc":"2.0"}'
```

3.4.2.1.10 eth_getTransactionByBlockHashAndIndex

简介

返回给定交易哈希和交易索引位置的交易

参数说明

参数	类型	说明
交易哈希	String	想要查询的交易的哈希值。
索引	String	编码为十六进制的交易索引位置。

返回值

- Object - 交易对象，如果未找到交易，则为 null。交易对象包含以下字段：
  - blockHash：此交易所在的区块的哈希值。当它是待处理的（Pending）日志时为 null
  - blockNumber：此交易所在的区块号。当它是待处理的（Pending）日志时为 null
  - from：发件人的地址
  - gas：发送方提供的gas，编码为十六进制
  - gasPrice：发件人提供的以 wei 为单位的gas价格，编码为十六进制
  - maxFeePerGas：交易中设置的每种gas的最高值
  - maxPriorityFeePerGas：交易中设置的最高优先级gas
  - hash：交易的哈希值
  - input：与交易一起发送的数据
  - nonce：发送方在此交易之前进行的交易数，编码为十六进制
  - to：接收方的地址。当它是合约创建交易时为 null
  - transactionIndex：从中创建日志的交易索引位置的整数。当它是待处理的（Pending）日志时为 null
  - value：以 wei 为单位的十六进制编码的转账数额
  - type：交易类型
  - accessList：交易计划访问的地址和存储密钥的列表
  - chainId：交易的链 ID（如果有）
  - v：签名的标准化 V 字段
  - r：签名的 R 字段
  - s：签名的 S 字段

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"eth_getTransactionByBlockHashAndIndex","params":
["0x81e807e7a6031d9f103eeee2a2edc5994c3432ee1e3227c66ff78eef30ea1dec","0x0"],"id":1,"jsonrpc":"2.0"}
'
```

3.4.2.1.11 eth_getTransactionByBlockNumberAndIndex

简介

返回给定区块号和交易索引位置的交易信息。该API所消耗的计算单元为18。

参数说明

参数	类型	说明
区块编号	String	想要查询的区块的十六进制的编号，或者是字符串"earliest"、"latest"、"pending"。
索引	String	编码为十六进制的交易索引位置。

返回值

- Object - 交易对象，如果未找到交易，则为 null。交易对象包含以下字段：
  - blockHash：此交易所在的区块的哈希值。当它是待处理的（Pending）日志时为 null
  - blockNumber：此交易所在的区块号。当它是待处理的（Pending）日志时为 null
  - from：发件人的地址
  - gas：发送方提供的gas，编码为十六进制
  - gasPrice：发件人提供的以 wei 为单位的gas价格，编码为十六进制
  - maxFeePerGas：交易中设置的每种gas的最高值
  - maxPriorityFeePerGas：交易中设置的最高优先级gas
  - hash：交易的哈希值
  - input：与交易一起发送的数据
  - nonce：发送方在此交易之前进行的交易数，编码为十六进制
  - to：接收方的地址。当它是合约创建交易时为 null
  - transactionIndex：从中创建日志的交易索引位置的整数。当它是待处理的（Pending）日志时为 null
  - value：以 wei 为单位的十六进制编码的转账数额
  - type：交易类型
  - accessList：交易计划访问的地址和存储密钥的列表
  - chainId：交易的链 ID（如果有）
  - v：签名的标准化 V 字段
  - r：签名的 R 字段
  - s：签名的 S 字段

请求样式

```
curl https://polygon-mainnet.shared-fullnode.bcs.ap-southeast-3.myhuaweicloud.com/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"eth_getTransactionByBlockNumberAndIndex","params":["0xc5043f",
"0x0"],"id":1,"jsonrpc":"2.0"}'
```

3.4.2.1.12 eth_getBlockTransactionCountByHash

简介

返回与给定块哈希匹配的区块的交易数。该API所消耗的计算单元为13。

参数说明

参数	类型	说明
区块哈希	String	想要查询的区块的哈希值。

返回值

以十六进制格式表示的所查询区块中的交易数。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"eth_getBlockTransactionCountByHash","params":
["0x81e807e7a6031d9f103eeee2a2edc5994c3432ee1e3227c66ff78eef30ea1dec"],"id":1,"jsonrpc":"2.0"}'
```

3.4.2.1.13 eth_getBlockTransactionCountByNumber

简介

返回与给定区块编号匹配的区块的交易数。该API所消耗的计算单元为12。

参数说明

参数	类型	说明
区块编号	String	想要查询的区块的十六进制的编号，或者是字符串"earliest"、"latest"、"pending"。

返回值

以十六进制格式表示的所查询区块中的交易数。

请求样式

```
curl https://polygon-mainnet.shared-fullnode.bcs.ap-southeast-3.myhuaweicloud.com/v1/<API-KEY> \
 -X POST \
 -H "Content-Type: application/json" \
 --data '{"method":"eth_getBlockTransactionCountByNumber","params":["0xc5043f"],"id":1,"jsonrpc":"2.0"}'
```

3.4.2.1.14 eth_getTransactionReceiptsByBlock

简介

返回给定区块号或哈希的所有交易收据。该API所消耗的计算单元为1100。

参数说明

参数	类型	说明
区块编号或者区块哈希	String	想要查询的区块的十六进制的编号，或者是字符串 "latest"，或是区块哈希值。

返回值

交易收据对象的数组，其中每个交易对象包含如下内容：

- Object - 交易收据对象，如果未找到交易收据，则为 null。交易收据对象包含以下字段：
  - blockHash：此交易所在的区块的哈希值
  - blockNumber：添加此交易的区块号，编码为十六进制
  - contractAddress：为创建合约创建的合约地址，如果并非合约创建则为空
  - cumulativeGasUsed：在区块中执行此交易时使用的总gas
  - from：源地址
  - gasUsed：仅此特定交易使用的gas
  - logs：生成此交易的日志对象数组
    - address：生成此日志的地址
    - topics：索引日志参数的 0 到 4 个 32 字节 DATA 的数组。在 Solidity 中，第一个主题是事件签名的哈希值（例如 Deposit(address, bytes32, uint256)），除非您使用匿名说明符声明事件
    - data：日志的 32 字节非索引参数
    - blockNumber：此日志所在的块号
    - transactionHash：从中创建此日志的交易的哈希。如果日志处于待处理（Pending）状态，则为 null
    - transactionIndex：从中创建此日志的交易索引位置。如果日志处于待处理（Pending）状态，则为 null

        - blockHash：此日志所在的块的哈希值
        - logIndex：编码为十六进制的块中对数索引位置的整数。如果日志处于待处理（Pending）状态，则为 null
        - removed：如果日志由于链重组而被删除，则为 true，如果它是有效的日志，则为 false。
    - logsBloom：用于检索相关日志的布隆过滤器
    - status：1（成功）或 0（失败），编码为十六进制
    - to：接收方的地址。当它是合约创建交易时为 null
    - transactionHash：交易的哈希值
    - transactionIndex：编码为十六进制的块中的交易索引位置
    - type：值的类型

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"eth_getTransactionReceiptsByBlock","params":["latest"],"id":1,"jsonrpc":"2.0"}'
```

3.4.2.1.15 eth_getTransactionReceipt

简介

通过交易哈希返回交易的收据。该API所消耗的计算单元为15。

参数说明

参数	类型	说明
交易哈希	String	想要查询的交易的哈希值。

返回值

- Object - 交易收据对象，如果未找到交易收据，则为 null。交易收据对象包含以下字段：
    - blockHash：此交易所在的区块的哈希值
    - blockNumber：添加此交易的区块号，编码为十六进制
    - contractAddress：为创建合约创建的合约地址，如果并非合约创建则为空
    - cumulativeGasUsed：在区块中执行此交易时使用的总gas
    - effectiveGasPrice：为每单位gas支付的总基本费用加上额外交易费
    - from：源地址
    - gasUsed：仅此特定交易使用的gas
    - logs：生成此交易的日志对象数组
        - address：生成此日志的地址

  - topics：索引日志参数的 0 到 4 个 32 字节 DATA 的数组。在 Solidity 中，第一个主题是事件签名的哈希值（例如 Deposit(address, bytes32, uint256)），除非您使用匿名说明符声明事件
  - data：日志的 32 字节非索引参数
  - blockNumber：此日志所在的块号
  - transactionHash：从中创建此日志的交易的哈希。如果日志处于待处理（Pending）状态，则为 null
  - transactionIndex：从中创建此日志的交易索引位置。如果日志处于待处理（Pending）状态，则为 null
  - blockHash：此日志所在的块的哈希值
  - logIndex：编码为十六进制的块中对数索引位置的整数。如果日志处于待处理（Pending）状态，则为 null
  - removed：如果日志由于链重组而被删除，则为 true，如果它是有效的日志，则为 false。
- logsBloom：用于检索相关日志的布隆过滤器
- status：1（成功）或 0（失败），编码为十六进制
- to：接收方的地址。当它是合约创建交易时为 null
- transactionHash：交易的哈希值
- transactionIndex：编码为十六进制的块中的交易索引位置
- type：值的类型

请求样式

```
curl https://polygon-mainnet.shared-fullnode.bcs.ap-southeast-3.myhuaweicloud.com/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"eth_getTransactionReceipt","params":
["0x6d755989f51032147484162c4dc3d6550552dbd8d3b094fe3c221bfa3c5942b2"],"id":1,"jsonrpc":"2.0"}'
```

3.4.2.1.16 eth_sendRawTransaction

简介

创建新的消息调用交易或为签名交易创建合约。该API所消耗的计算单元为300。

参数说明

参数	类型	说明
签名后的交易数据	String	使用您的私钥签名的交易。

返回值

交易哈希值，如果交易尚不可用，则为零哈希值。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"jsonrpc":"2.0","method":"eth_sendRawTransaction","params":["signed transaction"],"id":1}'
```

3.4.2.1.17 eth_call

简介

立即执行新的消息调用，而不在区块链上创建交易。该API所消耗的计算单元为20。

参数说明

包含交易的相关字段以及区块编号两部分。

参数	类型	说明
from	String	可选参数，发送交易的地址。
to	String	交易发送到的地址。
gas	Integer	可选参数，为交易执行提供的gas的整数。
gasPrice	Integer	可选参数，用于每个付费gas的gasPrice整数，编码为十六进制。
value	Integer	可选参数，与此交易一起发送的代币的数值，编码为十六进制。
data	String	可选参数，方法签名和编码参数的哈希值。有关更多信息，请参阅 Solidity 文档中的合约 ABI 描述。
区块编号	String	想要查询的区块的十六进制的编号，或者是字符串"earliest"、"latest"、"pending"。

返回值

执行合约方法的返回值。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"eth_call","params":
```

```
[{"from":null,"to":"0x6b175474e89094c44da98b954eedeac495271d0f","data":"0x70a082310000000000000000000000006E0d01A76C3Cf4288372a29124A26D4353EE51BE"}, "latest"],"id":1,"jsonrpc":"2.0"}'
```

3.4.2.1.18 eth_createAccessList

简介

基于给定的交易对象创建 EIP2930 类型 accessList。返回交易读取和写入的地址以及存储密钥列表，但是发送者帐户和预编译除外。该API所消耗的计算单元为300。

参数说明

包含交易的相关字段以及区块编号两部分。

参数	类型	说明
from	String	发送交易的地址
to	String	交易发送到的地址
gas	Integer	为交易执行提供的gas的整数
gasPrice	Integer	用于每个付费Gas的gasPrice整数，编码为十六进制
value	Integer	与此交易一起发送的代币的数值，编码为十六进制
data	String	方法签名和编码参数的哈希值。有关更多信息，请参阅 Solidity 文档中的合约 ABI 描述
区块编号	String	想要查询的区块的十六进制的编号，或者是字符串 "earliest"、"latest"、"pending"。

返回值

返回除发送者帐户和预编译之外的所有交易读取和写入的地址以及存储密钥列表，以及添加访问列表时消耗的估计gas。

- accessList: 具有以下字段的对象列表:
  - address: 交易要访问的地址。
  - storageKeys: 交易要访问的存储密钥。
- gasUsed: 十六进制字符串，表示交易的大致gas成本（ 如果包含访问列表 ）。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
```



```
  -H "Content-Type: application/json" \
  -d '{"method":"eth_createAccessList","params":[{"from":
"0xaeA8F8f781326bfE6A7683C2BD48Dd6AA4d3Ba63", "data": "0x608060806080608155"},
"pending"],"id":1,"jsonrpc":"2.0"}'
```

3.4.2.1.19 eth_estimateGas

简介

返回给定交易的所消耗的Gas的估计值。该API所消耗的计算单元为1000。

参数说明

与 eth_call 的参数一致，但所有属性都是可选的。 如果没有指定Gas限制，geth 将使用来自待处理区块的区块Gas限制作为上限。 因此，当所需Gas数量高于待处理区块的Gas限制时，返回的估算值可能不足以执行调用/交易。

参数	类型	说明
from	String	发送交易的地址。
to	String	交易发送到的地址。
gas	Integer	为交易执行提供的gas的整数。
gasPrice	Integer	用于每个付费gas的gasPrice整数，编码为十六进制。
value	Integer	与此交易一起发送的代币的数值，编码为十六进制。
data	String	方法签名和编码参数的哈希值。有关更多信息，请参阅 Solidity 文档中的合约 ABI 描述。
区块编号	String	想要查询的区块的十六进制的编号，或者是字符串 "earliest"、"latest"、"pending"。

返回值

交易所消耗Gas的预计数量。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"eth_estimateGas","params":
[{"from":"0x8D97689C9818892B700e27F316cc3E41e17fBeb9","to":"0xd3CdA913deB6f67967B99D67aCDFa1
712C293601","value":"0x186a0"}],"id":1,"jsonrpc":"2.0"}'
```

3.4.2.1.20 eth_feeHistory

简介

返回历史消耗的Gas信息的集合。该API所消耗的计算单元为17。

参数说明

参数	类型	说明
区块数量	String/Integer	请求范围内的块数。在单个查询中可以请求 1 到 1024 个块。如果不是所有块都可用，它将返回小于请求的范围。
最新区块编号	String	想要查询的区块的十六进制的编号，或者是字符串 "earliest"、"latest"、"pending"。
奖励百分位数	Integer	可选参数，单调递增的百分位值列表，从每个区块的每种 gas 的有效优先费中采样，按升序排列，并按所使用的 gas 进行加权。

返回值

- oldestBlock：以十六进制数表示的返回范围中最早的区块编号。
- baseFeePerGas：数组，内容为每个 gas 的一系列区块基本费用，包括额外的区块值。额外的值是返回范围中最新块之后的下一个块。对于EIP-1559之前创建的块返回零。
- gasUsedRatio：数组，内容为每个区块gas使用比率。计算方式为gasUsed和gasLimit的比率。
- reward：来自单个区块的每个Gas数据点的有效优先费数组。如果块为空，则返回所有零。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  -d '{"id": 1, "jsonrpc": "2.0", "method": "eth_feeHistory", "params": ["0x5", "latest", [20,30]] }'
```

3.4.2.1.21 eth_maxPriorityFeePerGas

简介

返回每个Gas的费用，这是您可以支付多少优先费用或“小费”的估计，以获得当前区块中包含的交易。该API所消耗的计算单元为13。

参数说明

此方法不接受任何参数。

返回值

十六进制代码编码的当前区块中包含交易的每项 gas 费用。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
   -X POST \
   -H "Content-Type: application/json" \
   -d '{"jsonrpc":"2.0","method":"eth_maxPriorityFeePerGas","id":1}'
```

3.4.2.1.22 eth_gasPrice

简介

返回当前的Gas价格（以 wei 为单位）。该API所消耗的计算单元为13。

参数说明

此方法不接受任何参数。

返回值

以十六进制表示的当前Gas的价格，以 wei 为单位。

请求样式

```
curl https://polygon-mainnet.shared-fullnode.bcs.ap-southeast-3.myhuaweicloud.com/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  -d '{"jsonrpc":"2.0","method":"eth_gasPrice","params": [],"id":1}'
```

3.4.2.1.23 eth_getBalance

简介

返回给定地址的帐户余额。该API所消耗的计算单元为15。

参数说明

参数	类型	说明
地址	String	表示用于检查余额的地址
区块编号	String	想要查询的区块的十六进制的编号，或者是字符串"earliest"、"latest"、"pending"。

返回值

以十六进制编码的给定地址帐户中当前余额，以wei为单位。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  -d '{"jsonrpc":"2.0","method":"eth_getBalance","params":
["0xc94770007dda54cF92009BFF0dE90c06F603a09f", "latest"],"id":1}'
```

3.4.2.1.24 eth_getRootHash

简介

返回指定区块范围的根哈希。该API所消耗的计算单元为30。

参数说明

参数	类型	说明
开始区块	Integer	开始区块的区块号。
结束区块	Integer	结束区块的区块号。

返回值

指定区块范围的根哈希。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  -d '{"jsonrpc":"2.0","method":"eth_getRootHash","params":[1000, 1032], "id":1}'
```

3.4.2.1.25 eth_subscribe

简介

为特定事件创建新订阅。节点返回订阅 ID。对于与订阅匹配的每个事件，将发送包含相关数据的通知以及订阅 ID。该API所消耗的计算单元为10。

参数说明

参数	类型	说明
事件类型	String	指定要侦听的事件类型。

参数	类型	说明
可选参数	String	要包含的可选参数，用于描述要侦听的事件类型，包括newHeads、newPendingTransactions、logs。

返回值

当订阅处于活动状态时，您将收到格式化为如下对象的事件：

事件对象：

- jsonrpc：始终为“2.0”。
- method：始终“eth_subscription”。
- params：具有以下字段的对象：
  - subscription：创建此订阅的调用返回的订阅 ID。此 ID 将附加到所有收到的事件，也可用于使用eth_unsubscribe。
  - result：内容因事件类型而异的对象。

请求样式

```
wscat -c wss://your-http-endpoint/v1/<API-KEY> -x '{"jsonrpc":"2.0", "id": 1, "method": "eth_subscribe", "params": ["logs"]}'
```

3.4.2.1.26 eth_unsubscribe

简介

通过使用订阅 ID 调用此方法来取消订阅。它返回一个布尔值，指示订阅已成功取消。该API所消耗的计算单元为10。

参数说明

参数	类型	说明
订阅ID	String	要取消订阅的订阅的 ID。

返回值

如果订阅已成功取消，返回True，否则返回False。

请求样式

```
wscat -c wss://your-http-endpoint/v1/<API-KEY> -x '{"jsonrpc":"2.0", "id": 1, "method": "eth_unsubscribe", "params": ["0x9cef478923ff08bf67fde6c64013158d"]}'
```

3.4.2.1.27 eth_getStorageAt

简介

返回给定地址的存储位置的值。该API所消耗的计算单元为15。

参数说明

参数	类型	说明
地址	String	表示存储地址（20 字节）的字符串。
存储位置	String	存储位置的十六进制代码。
区块编号	String	想要查询的区块的十六进制的编号，或者是字符串"earliest"、"latest"、"pending"。

返回值

所提供存储位置的值。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  -d '{"jsonrpc":"2.0","method":"eth_getStorageAt","params":
["0x295a70b2de5e3953354a6a8344e616ed314d7251",
"0x6661e9d6d8b923d5bbaab1b96e1dd51ff6ea2a93520fdc9eb75d059238b8c5e9", "0x65a8db"],"id":1}'
```

3.4.2.1.28 eth_accounts

简介

返回客户端拥有的地址列表。该API所消耗的计算单元为13。

参数说明

此方法不接受任何参数。

返回值

数组，包含客户端拥有的地址的十六进制编码的字符串。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
    -X POST \
    -H "Content-Type: application/json" \
    -d '{"jsonrpc":"2.0","method":"eth_accounts","params":[],"id":1}'
```

3.4.2.1.29 eth_getCode

简介

返回给定地址处智能合约的已编译字节代码（如果有）。该API所消耗的计算单元为40。

参数说明

参数	类型	说明
地址	String	表示存储地址（20 字节）的字符串，从中获取编译后的字节码。
区块编码	String	想要查询的区块的十六进制的编号，或者是字符串"earliest"、"latest"、"pending"。

返回值

给定地址处智能合约的已编译字节代码。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  -d '{"jsonrpc":"2.0","method":"eth_getCode","params":
["0x06012c8cf97bead5deae237070f9587f8e7a266d", "0x65a8db"],"id":1}'
```

3.4.2.1.30 eth_getProof

简介

返回指定账户的账户和存储值，包括 Merkle 证明。该API所消耗的计算单元为40。

参数说明

参数	类型	说明
账户地址	String	表示存储地址（20 字节）的字符串，从中获取编译后的字节码。
存储键	Array	要验证和包含的 32 字节存储键值的数组。
区块编号	String	想要查询的区块的十六进制的编号，或者是字符串"earliest"、"latest"、"pending"。

返回值

- address：与账户相关的地址。
- accountProof：RLP 序列化 MerkleTree-Nodes 的数组，从 stateRoot-Node 开始，遵循 SHA3（地址）的路径作为键。
- balance：当前余额的十六进制，以 wei 为单位。
- codeHash：账户代码的 32 字节哈希值。
- nonce：账户的随机数。
- storageHash: 32 字节。StorageRoot 的 SHA3。所有存储都将从这里开始提供 Merkle 证明rootHash。
- storageProof：请求的存储条目数组。每个条目都是一个具有以下属性的对象：
  - key：请求的存储密钥。
  - value：存储值。
  - proof：RLP 序列化 MerkleTree-Nodes 的数组，从 storageHash-Node 开始，遵循 SHA3（密钥）的路径作为路径。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  -d '{"jsonrpc": "2.0","method": "eth_getProof","id": 1,"params":
["0x7F0d15C7FAae65896648C8273B6d7E43f58Fa842",
["0x56e81f171bcc55a6ff8345e692c0f86e5b48e01b996cadc001622fb5e363b421"], "latest"]}'
```

3.4.2.1.31 eth_getLogs

简介

返回与给定过滤器对象匹配的所有日志的数组。该API所消耗的计算单元为75。

参数说明

参数	类型	说明
address	String	可选参数，合约地址（20字节）或日志应源自的地址列表。
fromBlock	String	可选参数，默认为“latest”，十六进制区块号，或字符串latest,earliest或pending。
toBlock	String	可选参数，默认为“latest”，十六进制区块号，或字符串latest,earliest或pending。

参数	类型	说明
topics	String	可选参数，32 字节 DATA 主题数组。主题与顺序相关。
blockhash	String	可选参数，将返回的日志限制为 32 字节哈希中引用的单个块blockHash。使用blockHash相当于设置fromBlock和 toBlock中引用的区块均为blockHash所对应的区块。如果blockHash出现在过滤条件中，则fromBlock和toBlock都不能够被设置。

返回值

日志对象数组，如果自上次轮询以来没有任何更改，则为空数组。日志对象包含以下键及其值：

- removed：若日志由于链重组而被删除，则返回true。如果它是有效的日志，则返回false。
- logIndex：块中日志索引位置的十六进制。当它是待处理（Pending）日志时返回NULL。
- transactionIndex：创建日志的事务索引位置的十六进制。当它是待处理（Pending）日志时返回NULL。
- transactionHash: 32 字节。创建此日志的事务的哈希值。当它是待处理（Pending）日志时返回NULL。
- blockHash: 32 字节。该日志所在块的哈希值，当它是待处理（Pending）日志时返回NULL。
- blockNumber：该日志所在的块号，当它是待处理（Pending）日志时返回NULL。
- address: 20 字节。该日志的来源地址。
- data：包含一个或多个 32 字节非索引日志参数。
- topics：包含 0 到 4 个索引日志参数的数组，每个 32 字节。在 Solidity 中，第一个主题是事件签名的哈希值（例如 Deposit(address,bytes32,uint256)），除非您使用匿名说明符声明事件。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  -d '{"jsonrpc":"2.0","method":"eth_getLogs","params":[{"blockHash":
"0x7c5a35e9cb3e8ae0e221ab470abae9d446c3a5626ce6689fc777dcffcab52c70", "topics":
["0x241ea03ca20251805084d27d4440371c34a0b85ff108f6bb5611248f73818b80"]}],"id":74}'
```

3.4.2.1.32 eth_getFilterChanges

简介

过滤器的轮询方法，返回自上次轮询以来发生的日志数组。过滤器必须通过调用eth_newFilter、eth_newBlockFilter、eth_newPendingTransactionFilter来创建。该API所消耗的计算单元为12。

参数说明

参数	类型	说明
过滤器ID	String	表示过滤器 ID 的字符串。

返回值

- log object array：（数组）日志对象数组，如果自上次轮询以来没有任何更改，则为空数组。
- 对于使用eth_newBlockFilter返回值创建的过滤器，返回值是块哈希（32 字节），例如["0x3454645634534..."]。
- 对于使用eth_newFilter日志创建的过滤器，对象具有以下参数：
  - address: 该日志的来源地址。
  - blockHash:该日志所在块的哈希值。当它是待处理（Pending）日志时返回NULL。
  - blockNumber：该日志所在的块号。当它是待处理（Pending）日志时返回NULL。
  - data： 包含日志的非索引参数。
  - logIndex：块中日志索引位置的十六进制。当它是待处理（Pending）日志时返回NULL。
  - removed：若日志由于链重组而被删除，则返回true。如果它是有效的日志，则返回false。
  - topics：数据数组。索引日志参数的 0 到 4 个 32 字节 DATA 的数组。在 Solidity 中，第一个topic是事件签名的哈希值（例如 Deposit(address,bytes32,uint256)），除非您使用匿名说明符声明事件。
  - transactionHash: 32 字节。创建此日志的事务的哈希值。当它是待处理（Pending）日志时返回NULL。
  - transactionIndex：创建此日志的事务索引位置的十六进制。当它是待处理（Pending）日志时返回NULL。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
   -X POST \
   -H "Content-Type: application/json" \
   -d '{"jsonrpc":"2.0","method":"eth_getFilterChanges","params":["0x16"],"id":73}'
```

3.4.2.1.33 eth_getFilterLogs

简介

返回与给定过滤器 ID 匹配的所有日志的数组。该API所消耗的计算单元为500。

参数说明

参数	类型	说明
address	String	可选参数，合约地址（20字节）或日志应源自的地址列表。
fromBlock	String	可选参数，默认为“latest”，十六进制区块号，或字符串latest，earliest或pending。
toBlock	String	可选参数，默认为“latest”，十六进制区块号，或字符串latest，earliest或pending。
topics	String	可选参数，32 字节 DATA 主题数组。主题与顺序相关。
blockhash	String	可选参数，将返回的日志限制为 32 字节哈希中引用的单个块blockHash。使用blockHash相当于设置fromBlock和 toBlock中引用的区块均为blockHash所对应的区块。如果blockHash出现在过滤条件中，则fromBlock和toBlock都不能够被设置。

返回值

- log 对象数组：与过滤器匹配的日志对象数组。对于自上次轮询以来发生的日志数组，请使用eth_getFilterChanges。日志对象包含以下键及其值：
  - address: 该日志的来源地址。
  - blockHash：该日志所在块的哈希值。当它是待处理（Pending）日志时返回NULL。
  - blockNumber：该日志所在的块号。当它是待处理（Pending）日志时返回NULL。
  - data：包含日志的非索引参数。
  - logIndex：块中日志索引位置的十六进制。当它是待处理（Pending）日志时返回NULL。
  - removed：若日志由于链重组而被删除，则返回true。如果它是有效的日志，则返回false。
  - topics：数据数组。索引日志参数的 0 到 4 个 32 字节 DATA 的数组。在 Solidity 中，第一个topic是事件签名的哈希值（例如 Deposit(address,bytes32,uint256)），除非您使用匿名说明符声明事件。
  - transactionHash: 创建此日志的事务的哈希值。当它是待处理（Pending）日志时返回NULL。
  - transactionIndex：创建此日志的事务索引位置的十六进制。当它是待处理（Pending）日志时返回NULL。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
   -X POST \
   -H "Content-Type: application/json" \
   -d '{"jsonrpc":"2.0","method":"eth_getFilterLogs","params":["0x16"],"id":74}'
```

3.4.2.1.34 eth_newBlockFilter

简介

在节点中创建一个过滤器，以在新区块到达时发出通知。该API所消耗的计算单元为24。

参数说明

此方法不接受任何参数。

返回值

十六进制表示的新创建的过滤器ID。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
   -X POST \
   -H "Content-Type: application/json" \
   -d '{"jsonrpc":"2.0","method":"eth_newBlockFilter","params":[],"id":73}'
```

3.4.2.1.35 eth_newFilter

简介

根据给定的过滤器选项创建过滤器对象，以在状态更改（日志）时发出通知。该API所消耗的计算单元为17。

参数说明

参数	类型	说明
address	String	可选参数，合约地址（20字节）或日志应源自的地址列表。
fromBlock	String	可选参数，默认为“latest”，十六进制区块号，或字符串latest，earliest或pending。
toBlock	String	可选参数，默认为“latest”，十六进制区块号，或字符串latest，earliest或pending。

参数	类型	说明
topics	String	可选参数，32 字节 DATA 主题数组。主题与顺序相关。

返回值

十六进制表示的新创建的过滤器ID。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
   -X POST \
   -H "Content-Type: application/json" \
   -d '{"jsonrpc":"2.0","method":"eth_newFilter","params":[{"topics":
["0xddf252ad1be2c89b69c2b068fc378daa952ba7f163c4a11628f55a4df523b3ef"]}],"id":73}'
```

3.4.2.1.36 eth_newPendingTransactionFilter

简介

在节点中创建一个过滤器，以在新的待处理事务到达 Polygon 时发出通知。该API所消耗的计算单元为24。

参数说明

此方法不接受任何参数。

返回值

十六进制表示的新创建的过滤器ID。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
   -X POST \
   -H "Content-Type: application/json" \
   -d '{"jsonrpc":"2.0","method":"eth_newPendingTransactionFilter","params":[],"id":73}'
```

3.4.2.1.37 eth_uninstallFilter

简介

卸载具有给定 id 的过滤器。当不再需要观察时调用。此外，如果一段时间内没有通过eth_getFilterChanges请求过滤器，过滤器就会超时。该API所消耗的计算单元为13。

参数说明

参数	类型	说明
过滤器ID	String	要卸载的过滤器ID的字符串。

返回值

如果过滤器已成功卸载，返回true，否则false。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
   -X POST \
   -H "Content-Type: application/json" \
   -d '{"jsonrpc":"2.0","method":"eth_uninstallFilter","params":["0xb"],"id":73}'
```

3.4.2.1.38 eth_chainId

简介

返回当前配置的链 ID。该API所消耗的计算单元为1。

参数说明

此方法不接受任何参数。

返回值

十六进制表示的链 ID。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
   -X POST \
   -H "Content-Type: application/json" \
   -d '{"jsonrpc":"2.0","method":"eth_chainId","params": [],"id":1}'
```

3.4.2.1.39 web3_sha3

简介

返回给定数据的 Keccak-256 编码结果（不是标准化 SHA3-256）。该API所消耗的计算单元为13。

参数说明

参数	类型	说明
数据	String	需要转换的数据

返回值

给定输入的SHA3编码后的结果。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
   -X POST \
```

```
-H "Content-Type: application/json" \
-d '{"jsonrpc":"2.0","method":"web3_sha3","params": ["0x68656c6c6f20776f726c64"],"id":64}'
```

3.4.2.1.40 web3_clientVersion

简介

返回当前客户端的版本。该API所消耗的计算单元为13。

参数说明

此方法不接受任何参数。

返回值

客户端的版本。

3.4.2.2 Polygon JSON-RPC API

3.4.2.2.1 bor_getAuthor

简介

返回指定块的作者。该API所消耗的计算单元为13。

参数说明

参数	类型	说明
区块编号	String	十六进制的区块编号，或者是字符串 "latest"。

返回值

区块作者的地址。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
-X POST \
-H "Content-Type: application/json" \
-d '{"jsonrpc":"2.0","method":"bor_getAuthor","params":["0x1000"], "id":1}'
```

3.4.2.2.2 bor_getCurrentProposer

简介

返回当前提议者的地址。该API所消耗的计算单元为13。

参数说明

此方法不接受任何参数。

返回值

当前提议者的地址。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
-X POST \
-H "Content-Type: application/json" \
-d '{"jsonrpc":"2.0","method":"bor_getCurrentProposer","params":[], "id":1}'
```

3.4.2.2.3 bor_getCurrentValidators

简介

返回当前验证者列表。该API所消耗的计算单元为19。

参数说明

此方法不接受任何参数。

返回值

当前验证者列表。每个验证者包含以下字段：

- ID：验证者ID
- accum：验证者的提议者优先级
- power：验证人的投票权
- signer: 验证人地址

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
-X POST \
-H "Content-Type: application/json" \
-d '{"jsonrpc":"2.0","method":"bor_getCurrentValidators","params":[], "id":1}'
```

3.4.2.2.4 bor_getRootHash

简介

返回指定区块范围的根哈希。该API所消耗的计算单元为13。

参数说明

参数	类型	说明
开始区块	int	开始区块的编号
结束区块	int	结束区块的编号

返回值

指定区块范围的根哈希。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
-X POST \
-H "Content-Type: application/json" \
-d '{"jsonrpc":"2.0","method":"bor_getRootHash","params":[1000, 1032], "id":1}'
```

3.4.2.2.5 bor_getSignersAtHash

简介

返回与指定块哈希匹配的块的所有签名者。该API所消耗的计算单元为16。

参数说明

参数	类型	说明
区块哈希	String	想要查询的区块的哈希值

返回值

指定块哈希的签名者数组。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
-X POST \
-H "Content-Type: application/json" \
-d '{"jsonrpc":"2.0","method":"bor_getSignersAtHash","params":
["0x29fa73e3da83ddac98f527254fe37002e052725a88904bac14f03e919e1e2876"], "id":1}'
```


4 Arbitrum

4.1 Arbitrum 介绍

Arbitrum是一个基于以太坊的第二层扩容方案，它可以提高以太坊的可扩展性、降低网络拥堵和交易费用，同时保持与以太坊虚拟机（EVM）的兼容性和安全性。Arbitrum采用了一种叫做Optimistic Rollup的技术，它可以将大量的交易打包成一个区块，并提交到以太坊主链上，只有在出现争议时才需要验证区块的正确性。这样可以大幅减少对以太坊主链的资源占用，提高交易吞吐量和速度。

Arbitrum官方链接：开发者中心，白皮书

用户可以通过使用华为云公链节点引擎，来提升区块链使用与开发的效率，增强其稳定性与私密性。**华为云将永远不会收集用户的区块链地址。**

说明

- **支持网络**
  - Arbitrum：HTTP，WebSocket
- **Arbitrum API列表**

4.2 HTTP 请求示例

4.2.1 使用 cURL 发送 HTTP API 请求

Request example:

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"eth_getBlockByNumber","params":["0xc5043f",false],"id":1,"jsonrpc":"2.0"}'
```

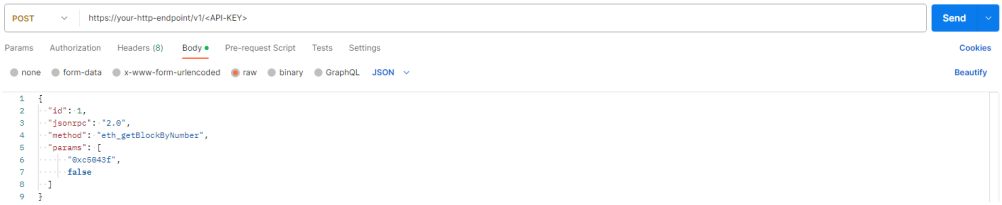
Response example:

```
{
    "jsonrpc": "2.0",
    "id": 1,
    "result": {
        "difficulty": "0x0",
```

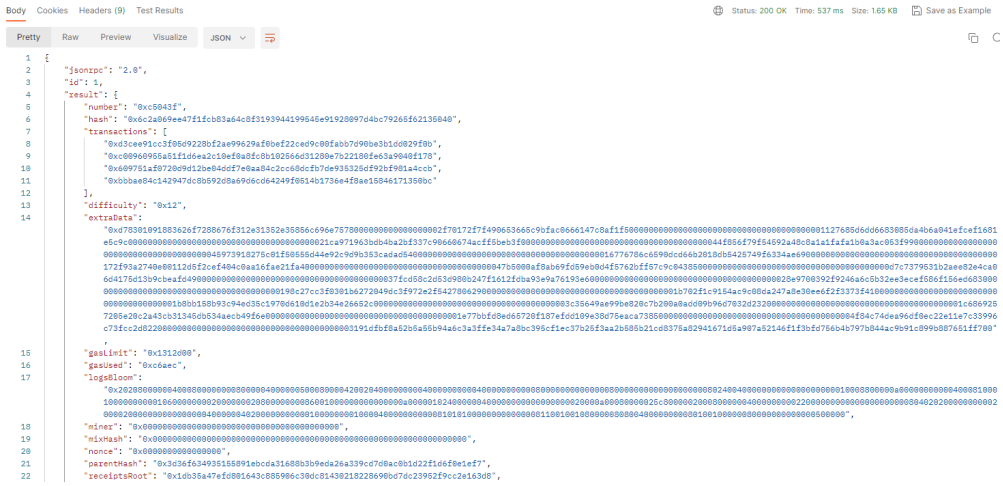
```
        "extraData": "0x",
        "gasLimit": "0xbc87657",
        "gasUsed": "0x3daa0",
        "hash": "0x8f9ecad637559914862de6821bd352d6ac7744d130085d4c5b3d821786aab3ac",
        "l1BlockNumber": "0xe2733e",
        "logsBloom":
"0x0008000000000000000800000000001000000000000000000002000000000000000000000000
0080000000000000004400000000000040000000000008000000000000000000080000000000
0000000000000000000002000000000000000000800000000000000000000000000000000000
0000000000040000000000800000000000400800400000000000000800000000000000000000
0000000000000000000000000000000000040000000000000000040000000000000004000000
0000000000000000000000000000000000040000000000000000000000000800000000001000
0400000000020000000000000000000000020000000",
        "miner": "0x0000000000000000000000000000000000000000",
        "mixHash":
"0x0000000000000000000000000000000000000000000000000000000000000000",
        "nonce": "0x0000000000000000",
        "number": "0xc5043f",
        "parentHash":
"0xf1b5d6a7a45df869b2eef85541584c69bbeb7a58f77e557d4898de2b464d5715",
        "receiptsRoot":
"0x17f8ad0067aff1dbb35d5100b1f131838564b3e980e2b8a2674b8d5362d12e97",
        "sha3Uncles":
"0x1dcc4de8dec75d7aab85b567b6ccd41ad312451b948a7413f0a142fd40d49347",
        "size": "0x367",
        "stateRoot":
"0x0000000000000000000000000000000000000000000000000000000000000000",
        "timestamp": "0x628dde24",
        "totalDifficulty": "0x0",
        "transactions": [
          {
            "blockHash":
"0x8f9ecad637559914862de6821bd352d6ac7744d130085d4c5b3d821786aab3ac",
            "blockNumber": "0xc5043f",
            "from": "0x1eedcd7c2334463c51b4af75c96b983be7ed4d39",
            "gas": "0xd5d2f",
            "gasPrice": "0x14c30896",
            "hash":
"0x39229c9c973b7675086e4404c30fd23b4130bdb33cef8f799ac0450e2489ba08",
            "input":
"0x0a0e28d4000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000004000000000000000000
0000000000000000000000000000000000000000000046a000000000000000000000000000000000
0000000000000000000000000000004690000000000000000000000000000000000000000000000000
0000000000000046b000000000000000000000000000000000000000000000000000000000000000
5ea2",
            "nonce": "0x324",
            "to": "0x737eaf14061fe68f04ff4ca8205acf538555fcc8",
            "transactionIndex": "0x0",
            "value": "0x0",
            "type": "0x78",
            "v": "0x14985",
            "r": "0x64445cd16ea28f39f5b94d3730147f4f0fde0dd549adfc9c80c8948a3679e878",
            "s": "0x39851deff11a344c3fd67c022bcaf9711a5c8eae60f81d47a977253df9e95e39"
          }
        ],
        "transactionsRoot":
"0xc4b4888ec249430e95f700432d21a3042e5dd2572e68771dc61f73e1fbbd9900",
        "uncles": []
    }
}
```


4.2.2 使用 post-man 发送 HTTP API 请求

Request example:



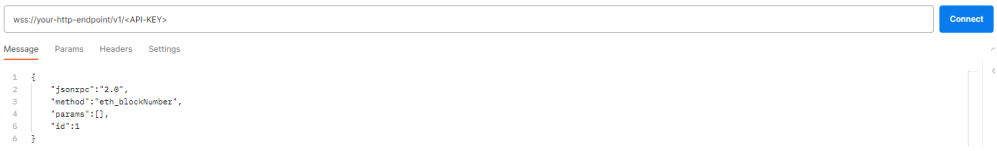
Response example:



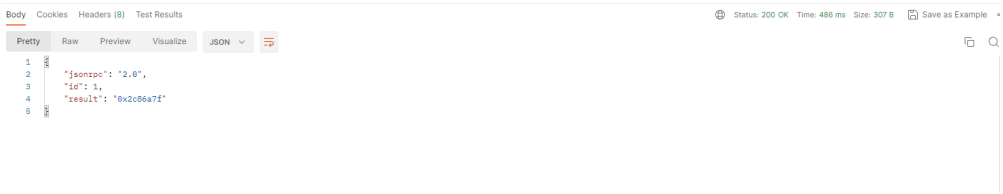
4.3 WebSocket 请求示例

4.3.1 使用 post-man 发送 JSON-RPC API 请求

Request example:



Response example:



4.4 Arbitrum API 列表

4.4.1 共享版

4.4.1.1 Ethereum JSON-RPC API

4.4.1.1.1 eth_blocknumber

简介

返回区块链的最新区块号。该API所消耗的计算单元为13。

参数说明

此方法不接受任何参数。

返回值

十六进制编码的最新区块号。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"eth_blockNumber","params":[],"id":1,"jsonrpc":"2.0"}'
```

4.4.1.1.2 eth_getBlockByNumber

简介

返回与给定的区块号匹配的区块信息。该API所消耗的计算单元为21。

参数说明

参数	类型	说明
区块编号	String	十六进制的区块编号，或者是字符串 "earliest"、"latest"、"pending"。
交易详细信息标志	Bool	当此值为 true 时，该方法返回完整的交易对象，否则，它仅返回交易的哈希。

返回值

- Object - 区块对象，如果未找到区块，则为 null。区块对象包含以下字段：
  - number: 编码为十六进制的请求块的块号。如果是一个处于待处理（ Pending ）状态的区块，则为空。
  - hash: 区块的哈希。如果是一个处于待处理（ Pending ）状态的区块，则为空。

- parentHash: 父区块的哈希。
- nonce: 生成的工作量证明的哈希值。如果是一个处于待处理（ Pending ）状态的区块，则为空。
- sha3Uncles: 区块中叔区块数据的 SHA3。
- logsBloom: 区块日志的布隆过滤器。如果是一个处于待处理（ Pending ）状态的区块，则为空。
- transactionsRoot: 区块中交易树的根。
- stateRoot： 区块的最终状态树的根。
- receiptsRoot： 区块的收据树的根。
- miner： 获得采矿奖励的受益人的地址。
- difficulty：此区块的难度。
- totalDifficulty：直到这个区块时，链的总难度。
- extraData：此区块的“额外数据”字段。
- size：此区块的大小（以字节为单位）。
- gasLimit：此区块中允许的最大gas。
- gasUsed：此区块中所有交易的总使用gas。
- timestamp：整理区块时的 unix 时间戳。
- transactions：交易对象的数组，或 32 字节的交易哈希，具体取决于最后一个给定的参数。
- uncles：叔区块哈希数组。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"eth_getBlockByNumber","params":["0xc5043f",false],"id":1,"jsonrpc":"2.0"}'
```

4.4.1.1.3 eth_getUncleByBlockNumberAndIndex

简介

按区块编号和叔区块索引位置返回有关叔区块的信息。该API所消耗的计算单元为15。

参数说明

参数	类型	说明
区块编号或标签	String	想要查询的区块的十六进制的编号，或者是字符串"earliest"、"latest"、"pending"。
叔区块的索引位置	String	叔区块的十六进制的区块编号。

返回值

- Object - 区块对象，如果未找到区块，则为 null。区块对象包含以下字段：
  - number: 编码为十六进制的请求块的块号。如果是一个处于待处理（Pending）状态的区块，则为空。
  - hash: 区块的哈希。如果是一个处于待处理（Pending）状态的区块，则为空。
  - parentHash: 父区块的哈希。
  - nonce: 生成的工作量证明的哈希值。如果是一个处于待处理（Pending）状态的区块，则为空。
  - sha3Uncles: 区块中叔区块数据的 SHA3。
  - logsBloom: 区块日志的布隆过滤器。如果是一个处于待处理（Pending）状态的区块，则为空。
  - transactionsRoot: 区块中交易树的根。
  - stateRoot： 区块的最终状态树的根。
  - receiptsRoot： 区块的收据树的根。
  - miner： 获得采矿奖励的受益人的地址。
  - difficulty：此区块的难度。
  - totalDifficulty： 直到这个区块时，链的总难度。
  - extraData：此区块的“额外数据”字段。
  - size：此区块的大小（以字节为单位）。
  - gasLimit：此区块中允许的最大gas。
  - gasUsed：此区块中所有交易的总使用gas。
  - timestamp：整理区块时的 unix 时间戳。
  - transactions：交易对象的数组，或 32 字节的交易哈希，具体取决于最后一个给定的参数。
  - uncles：叔区块哈希数组。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"eth_getUncleByBlockNumberAndIndex","params":["latest","0x0"],"id":1,"jsonrpc":"2.0"}'
```

4.4.1.1.4 eth_getUncleByBlockHashAndIndex

简介

按区块哈希和叔区块索引位置返回有关叔区块的信息。该API所消耗的计算单元为16。

参数说明

参数	类型	说明
区块哈希	String	想要查询的区块的哈希值。

参数	类型	说明
叔区块的索引位置	String	叔区块的十六进制的区块编号。

返回值

Object - 区块对象，如果未找到区块，则为 null。区块对象包含以下字段：

- number: 编码为十六进制的请求块的块号。如果是一个处于待处理（Pending）状态的区块，则为空。
- hash: 区块的哈希。如果是一个处于待处理（Pending）状态的区块，则为空。
- parentHash: 父区块的哈希。
- nonce: 生成的工作量证明的哈希值。如果是一个处于待处理（Pending）状态的区块，则为空。
- sha3Uncles: 区块中叔区块数据的 SHA3。
- logsBloom: 区块日志的布隆过滤器。如果是一个处于待处理（Pending）状态的区块，则为空。
- transactionsRoot: 区块中交易树的根。
- stateRoot： 区块的最终状态树的根。
- receiptsRoot： 区块的收据树的根。
- miner： 获得采矿奖励的受益人的地址。
- difficulty：此区块的难度。
- totalDifficulty： 直到这个区块时，链的总难度。
- extraData：此区块的“额外数据”字段。
- size：此区块的大小（以字节为单位）。
- gasLimit：此区块中允许的最大gas。
- gasUsed：此区块中所有交易的总使用gas。
- timestamp：整理区块时的 unix 时间戳。
- transactions：交易对象的数组，或 32 字节的交易哈希，具体取决于最后一个给定的参数。
- uncles：叔区块哈希数组。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"eth_getUncleByBlockHashAndIndex","params":
[0xc6ef2fc5426d6ad6fd9e2a26abeab0aa2411b7ab17f30a99d3cb96aed1d1055b",
"0x0"],"id":1,"jsonrpc":"2.0"}'
```

4.4.1.1.5 eth_getUncleCountByBlockNumber

简介

返回与给定区块编号匹配的区块中叔区块的数量。该API所消耗的计算单元为16。

参数说明

参数	类型	说明
区块编号	String	想要查询的区块的十六进制的编号。

返回值

区块中叔区块的数量，以十六进制为编码。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"eth_getUncleCountByBlockNumber","params":["0xc5043f"],"id":1,"jsonrpc":"2.0"}'
```

4.4.1.1.6 eth_getUncleCountByBlockHash

简介

返回与给定区块哈希匹配的区块中叔区块的数量。该API所消耗的计算单元为16。

参数说明

参数	类型	说明
区块哈希	String	想要查询的区块的哈希值。

返回值

区块中叔区块的数量，以十六进制为编码。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"eth_getUncleCountByBlockHash","params":
["0x81e807e7a6031d9f103eeee2a2edc5994c3432ee1e3227c66ff78eef30ea1dec"],"id":1,"jsonrpc":"2.0"}'
```

4.4.1.1.7 eth_getBlockByHash

简介

返回与给定区块哈希匹配的区块的信息。该API所消耗的计算单元为21。

参数说明

参数	类型	说明
区块哈希	String	想要查询的区块的哈希值。
事务详细信息标志	Bool	当此值为 true 时，该方法返回完整的交易对象，否则，它仅返回交易的哈希。

返回值

Object - 区块对象，如果未找到区块，则为 null。区块对象包含以下字段：

- number: 编码为十六进制的请求块的块号。如果是一个处于待处理（Pending）状态的区块，则为空。
- hash: 区块的哈希。如果是一个处于待处理（Pending）状态的区块，则为空。
- parentHash: 父区块的哈希。
- nonce: 生成的工作量证明的哈希值。如果是一个处于待处理（Pending）状态的区块，则为空。
- sha3Uncles: 区块中叔区块数据的 SHA3。
- logsBloom: 区块日志的布隆过滤器。如果是一个处于待处理（Pending）状态的区块，则为空。
- transactionsRoot: 区块中交易树的根。
- stateRoot： 区块的最终状态树的根。
- receiptsRoot： 区块的收据树的根。
- miner： 获得采矿奖励的受益人的地址。
- difficulty：此区块的难度。
- totalDifficulty： 直到这个区块时，链的总难度。
- extraData：此区块的“额外数据”字段。
- size：此区块的大小（以字节为单位）。
- gasLimit：此区块中允许的最大gas。
- gasUsed：此区块中所有交易的总使用gas。
- timestamp：整理区块时的 unix 时间戳。
- transactions：交易对象的数组，或 32 字节的交易哈希，具体取决于最后一个给定的参数。
- uncles：叔区块哈希数组。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"eth_getBlockByHash","params":
["0x81e807e7a6031d9f103eeee2a2edc5994c3432ee1e3227c66ff78eef30ea1dec",false],"id":1,"jsonrpc":"2.0"}'
```

4.4.1.1.8 eth_getTransactionByHash

简介

根据交易哈希返回有关交易的信息。该API所消耗的计算单元为17。

参数说明

参数	类型	说明
交易哈希	String	要查询的交易的哈希值。

返回值

Object - 交易对象，如果未找到交易，则为 null。交易对象包含以下字段：

- blockHash：此交易所在的区块的哈希值。当它是待处理的（Pending）日志时为 null
- blockNumber：此交易所在的区块号。当它是待处理的（Pending）日志时为 null
- from：发件人的地址
- gas：发送方提供的gas，编码为十六进制
- gasPrice：发件人提供的 wei 格式的gas价格，编码为十六进制
- maxFeePerGas：交易中设置的每种gas的最高值
- maxPriorityFeePerGas：交易中设置的最高优先级gas
- hash：交易的哈希值
- input：与交易一起发送的数据
- nonce：发送方在此交易之前进行的交易数，编码为十六进制
- to：接收方的地址。当它是合约创建交易时为 null
- transactionIndex：从中创建日志的交易索引位置的整数。当它是待处理的（Pending）日志时为 null
- value：以 wei 为单位的十六进制编码的转账数额
- type：交易类型
- accessList：交易计划访问的地址和存储密钥的列表
- chainId：交易的链 ID（如果有）
- v：签名的标准化 V 字段
- r：签名的 R 字段
- s：签名的 S 字段

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"eth_getTransactionByHash","params":
["0xb142342a7fd70602b7a0ba3688a41bfcbb4fbc3490c252ca48af2594619d220c"],"id":1,"jsonrpc":"2.0"}'
```

4.4.1.1.9 eth_getTransactionCount

简介

返回从某一地址发送的交易数。该API所消耗的计算单元为26。

参数说明

参数	类型	说明
地址	String	要检查的交易计数的地址。
区块编号	String	想要查询的区块的十六进制的编号，或者是字符串"earliest"、"latest"、"pending"。

返回值

从地址发送的十六进制编码的交易数量

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"eth_getTransactionCount","params":
["0x8D97689C9818892B700e27F316cc3E41e17fBeb9", "latest"],"id":1,"jsonrpc":"2.0"}'
```

4.4.1.1.10 eth_getTransactionByBlockHashAndIndex

简介

返回给定交易哈希和交易索引位置的交易信息。该API所消耗的计算单元为16。

参数说明

参数	类型	说明
交易哈希	String	想要查询的交易的哈希值。
索引	String	编码为十六进制的交易索引位置。

返回值

Object - 交易对象，如果未找到交易，则为 null。交易对象包含以下字段：

- blockHash：此交易所在的区块的哈希值。当它是待处理的（Pending）日志时为null
- blockNumber：此交易所在的区块号。当它是待处理的（Pending）日志时为null

- from：发件人的地址
- gas：发送方提供的gas，编码为十六进制
- gasPrice：发件人提供的以 wei 为单位的gas价格，编码为十六进制
- maxFeePerGas：交易中设置的每种gas的最高值
- maxPriorityFeePerGas：交易中设置的最高优先级gas
- hash：交易的哈希值
- input：与交易一起发送的数据
- nonce：发送方在此交易之前进行的交易数，编码为十六进制
- to：接收方的地址。当它是合约创建交易时为 null
- transactionIndex：从中创建日志的交易索引位置的整数。当它是待处理的（Pending）日志时为 null
- value：以 wei 为单位的十六进制编码的转账数额
- type：交易类型
- accessList：交易计划访问的地址和存储密钥的列表
- chainId：交易的链 ID（如果有）
- v：签名的标准化 V 字段
- r：签名的 R 字段
- s：签名的 S 字段

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"eth_getTransactionByBlockHashAndIndex","params":
["0x81e807e7a6031d9f103eeee2a2edc5994c3432ee1e3227c66ff78eef30ea1dec","0x0"],"id":1,"jsonrpc":"2.0"}
'
```

4.4.1.1.11 eth_getTransactionByBlockNumberAndIndex

简介

返回给定区块号和交易索引位置的交易信息。该API所消耗的计算单元为17。

参数说明

参数	类型	说明
区块编号	String	想要查询的区块的十六进制的编号，或者是字符串"earliest"、"latest"、"pending"。
索引	String	编码为十六进制的交易索引位置。

返回值

Object - 交易对象，如果未找到交易，则为 null。交易对象包含以下字段：

- blockHash：此交易所在的区块的哈希值。当它是待处理的（Pending）日志时为 null
- blockNumber：此交易所在的区块号。当它是待处理的（Pending）日志时为 null
- from：发件人的地址
- gas：发送方提供的gas，编码为十六进制
- gasPrice：发件人提供的以 wei 为单位的gas价格，编码为十六进制
- maxFeePerGas：交易中设置的每种gas的最高值
- maxPriorityFeePerGas：交易中设置的最高优先级gas
- hash：交易的哈希值
- input：与交易一起发送的数据
- nonce：发送方在此交易之前进行的交易数，编码为十六进制
- to：接收方的地址。当它是合约创建交易时为 null
- transactionIndex：从中创建日志的交易索引位置的整数。当它是待处理的（Pending）日志时为 null
- value：以 wei 为单位的十六进制编码的转账数额
- type：交易类型
- accessList：交易计划访问的地址和存储密钥的列表
- chainId：交易的链 ID（如果有）
- v：签名的标准化 V 字段
- r：签名的 R 字段
- s：签名的 S 字段

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"eth_getTransactionByBlockNumberAndIndex","params":["0xc5043f",
"0x0"],"id":1,"jsonrpc":"2.0"}'
```

4.4.1.1.12 eth_getBlockTransactionCountByHash

简介

返回与给定块哈希匹配的区块的交易数。该API所消耗的计算单元为20。

参数说明

参数	类型	说明
区块哈希	String	想要查询的区块的哈希值。

返回值

以十六进制格式表示的所查询区块中的交易数。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"eth_getBlockTransactionCountByHash","params":
["0x81e807e7a6031d9f103eeee2a2edc5994c3432ee1e3227c66ff78eef30ea1dec"],"id":1,"jsonrpc":"2.0"}'
```

4.4.1.1.13 eth_getBlockTransactionCountByNumber

简介

返回与给定区块编号匹配的区块的交易数。该API所消耗的计算单元为20。

参数说明

参数	类型	说明
区块编号	String	想要查询的区块的十六进制的编号，或者是字符串"earliest"、"latest"、"pending"。

返回值

以十六进制格式表示的所查询区块中的交易数。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"eth_getBlockTransactionCountByNumber","params":["0xc5043f"],"id":1,"jsonrpc":"2.0"}'
```

4.4.1.1.14 eth_syncing

简介

返回当前的同步状态。该API所消耗的计算单元为1。

参数说明

此方法不接受任何参数。

返回值

交易收据对象的数组，其中每个交易对象包含如下内容：

返回值1：

Boolean - 若同步完成，返回false

返回值2：

Object - 若同步中，返回同步数据状态

- startingBlock：导入的开始区块号，编码为十六进制
- currentBlock：当前的区块号，与**eth_blockNumber**结果相同，编码为十六进制
- highestBlock：预估的最高区块号，编码为十六进制

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
   -X POST \
   -H "Content-Type: application/json" \
   -d '{"jsonrpc":"2.0","method":"eth_syncing","params": [],"id":1}'
```

4.4.1.1.15 eth_getTransactionReceipt

简介

通过交易哈希返回交易的收据。该API所消耗的计算单元为17。

参数说明

参数	类型	说明
交易哈希	String	想要查询的交易的哈希值。

返回值

Object - 交易收据对象，如果未找到交易收据，则为 null。交易收据对象包含以下字段：

- blockHash：此交易所在的区块的哈希值
- blockNumber：添加此交易的区块号，编码为十六进制
- contractAddress：为创建合约创建的合约地址，如果并非合约创建则为空
- cumulativeGasUsed：在区块中执行此交易时使用的总gas
- effectiveGasPrice：为每单位gas支付的总基本费用加上额外交易费
- from：源地址
- gasUsed：仅此特定交易使用的gas
- logs：生成此交易的日志对象数组
  - address：生成此日志的地址
  - topics：索引日志参数的 0 到 4 个 32 字节 DATA 的数组。在 Solidity 中，第一个主题是事件签名的哈希值（例如 Deposit(address, bytes32, uint256)），除非您使用匿名说明符声明事件
  - data：日志的 32 字节非索引参数
  - blockNumber：此日志所在的块号
  - transactionHash：从中创建此日志的交易的哈希。如果日志处于待处理（Pending）状态，则为 null

  - transactionIndex：从中创建此日志的交易索引位置。如果日志处于待处理（Pending）状态，则为 null
  - blockHash：此日志所在的块的哈希值
  - logIndex：编码为十六进制的块中对数索引位置的整数。如果日志处于待处理（Pending）状态，则为 null
  - removed：如果日志由于链重组而被删除，则为 true，如果它是有效的日志，则为 false。
- logsBloom：用于检索相关日志的布隆过滤器
- status：1（成功）或 0（失败），编码为十六进制
- to：接收方的地址。当它是合约创建交易时为 null
- transactionHash：交易的哈希值
- transactionIndex：编码为十六进制的块中的交易索引位置
- type：值的类型

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"eth_getTransactionReceipt","params":
["0x6d755989f51032147484162c4dc3d6550552dbd8d3b094fe3c221bfa3c5942b2"],"id":1,"jsonrpc":"2.0"}'
```

4.4.1.1.16 eth_sendRawTransaction

简介

创建新的消息调用交易或为签名交易创建合约。该API所消耗的计算单元为308。

参数说明

参数	类型	说明
签名后的交易数据	String	使用您的私钥签名的交易。

返回值

交易哈希值，如果交易尚不可用，则为零哈希值。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"jsonrpc":"2.0","method":"eth_sendRawTransaction","params":["signed transaction"],"id":1}'
```

4.4.1.1.17 eth_call

简介

立即执行新的消息调用，而不在区块链上创建交易。该API所消耗的计算单元为41。

参数说明

包含交易的相关字段以及区块编号两部分。

参数	类型	说明
from	String	可选参数，发送交易的地址。
to	String	交易发送到的地址。
gas	Integer	可选参数，为交易执行提供的gas的整数。
gasPrice	Integer	可选参数，用于每个付费gas的gasPrice整数，编码为十六进制。
value	Integer	可选参数，与此交易一起发送的代币的数值，编码为十六进制。
data	String	可选参数，方法签名和编码参数的哈希值。有关更多信息，请参阅 Solidity 文档中的合约 ABI 描述。
区块编号	String	想要查询的区块的十六进制的编号，或者是字符串"earliest"、"latest"、"pending"。

返回值

执行合约方法的返回值。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"eth_call","params":
[{"from":null,"to":"0x6b175474e89094c44da98b954eedeac495271d0f","data":"0x70a08231000000000000000000000000006E0d01A76C3Cf4288372a29124A26D4353EE51BE"}, "latest"],"id":1,"jsonrpc":"2.0"}'
```

4.4.1.1.18 eth_createAccessList

简介

基于给定的交易对象创建 EIP2930 类型 accessList。返回交易读取和写入的地址以及存储密钥列表，但是发送者帐户和预编译除外。该API所消耗的计算单元为44。

参数说明

包含交易的相关字段以及区块编号两部分。

参数	类型	说明
from	String	发送交易的地址
to	String	交易发送到的地址
gas	Integer	为交易执行提供的gas的整数
gasPrice	Integer	用于每个付费Gas的gasPrice整数，编码为十六进制
value	Integer	与此交易一起发送的代币的数值，编码为十六进制
data	String	方法签名和编码参数的哈希值。有关更多信息，请参阅 Solidity 文档中的合约 ABI 描述
区块编号	String	想要查询的区块的十六进制的编号，或者是字符串"earliest"、"latest"、"pending"。

返回值

返回除发送者帐户和预编译之外的所有交易读取和写入的地址以及存储密钥列表，以及添加访问列表时消耗的估计gas。

accessList: 具有以下字段的对象列表:

- address: 交易要访问的地址。
- storageKeys: 交易要访问的存储密钥。
- gasUsed: 十六进制字符串，表示交易的大致gas成本（ 如果包含访问列表 ）。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
    -X POST \
    -H "Content-Type: application/json" \
    -d '{"method":"eth_createAccessList","params":[{"from":
"0xaeA8F8f781326bfE6A7683C2BD48Dd6AA4d3Ba63", "data": "0x608060806080608155"},
"pending"],"id":1,"jsonrpc":"2.0"}'
```

4.4.1.1.19 eth_estimateGas

简介

返回给定交易的所消耗的Gas的估计值。该API所消耗的计算单元为87。

参数说明

与 eth_call 的参数一致，但所有属性都是可选的。 如果没有指定Gas限制，geth 将使用来自待处理区块的区块Gas限制作为上限。 因此，当所需Gas数量高于待处理区块的Gas限制时，返回的估算值可能不足以执行调用/交易。

参数	类型	说明
from	String	发送交易的地址。
to	String	交易发送到的地址。
gas	Integer	为交易执行提供的gas的整数。
gasPrice	Integer	用于每个付费gas的gasPrice整数，编码为十六进制。
value	Integer	与此交易一起发送的代币的数值，编码为十六进制。
data	String	方法签名和编码参数的哈希值。有关更多信息，请参阅 Solidity 文档中的合约 ABI 描述。
区块编号	String	想要查询的区块的十六进制的编号，或者是字符串"earliest"、"latest"、"pending"。

返回值

交易所消耗Gas的预计数量。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"eth_estimateGas","params":
[{"from":"0x8D97689C9818892B700e27F316cc3E41e17fBeb9","to":"0xd3CdA913deB6f67967B99D67aCDFa1712C293601","value":"0x186a0"}],"id":1,"jsonrpc":"2.0"}'
```

4.4.1.1.20 eth_feeHistory

简介

返回历史消耗的Gas信息的集合。该API所消耗的计算单元为32。

参数说明

参数	类型	说明
区块数量	String/Integer	请求范围内的块数。在单个查询中可以请求 1 到 1024 个块。如果不是所有块都可用，它将返回小于请求的范围。
最新区块编号	String	想要查询的区块的十六进制的编号，或者是字符串 "earliest"、"latest"、"pending"。
奖励百分位数	Integer	可选参数，单调递增的百分位值列表，从每个区块的每种 gas 的有效优先费中采样，按升序排列，并按所使用的 gas 进行加权。

返回值

- oldestBlock：以十六进制数表示的返回范围中最早的区块编号。
- baseFeePerGas：数组，内容为每个 gas 的一系列区块基本费用，包括额外的区块值。额外的值是返回范围中最新块之后的下一个块。对于EIP-1559之前创建的块返回零。
- gasUsedRatio：数组，内容为每个区块gas使用比率。计算方式为gasUsed和gasLimit的比率。
- reward：来自单个区块的每个Gas数据点的有效优先费数组。如果块为空，则返回所有零。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  -d '{"id": 1, "jsonrpc": "2.0", "method": "eth_feeHistory", "params": ["0x5", "latest", [20,30]] }'
```

4.4.1.1.21 eth_maxPriorityFeePerGas

简介

返回每个Gas的费用，这是您可以支付多少优先费用或“小费”的估计，以获得当前区块中包含的交易。该API所消耗的计算单元为13。

参数说明

此方法不接受任何参数。

返回值

十六进制代码编码的当前区块中包含交易的每项 gas 费用。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
    -X POST \
    -H "Content-Type: application/json" \
    -d '{"jsonrpc":"2.0","method":"eth_maxPriorityFeePerGas","id":1}'
```

4.4.1.1.22 eth_gasPrice

简介

返回当前的Gas价格（以 wei 为单位）。该API所消耗的计算单元为19。

参数说明

此方法不接受任何参数。

返回值

以十六进制表示的当前Gas的价格，以 wei 为单位。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  -d '{"jsonrpc":"2.0","method":"eth_gasPrice","params": [],"id":1}'
```

4.4.1.1.23 eth_getBalance

简介

返回给定地址的帐户余额。该API所消耗的计算单元为19。

参数说明

参数	类型	说明
地址	String	表示用于检查余额的地址
区块编号	String	想要查询的区块的十六进制的编号，或者是字符串"earliest"、"latest"、"pending"。

返回值

以十六进制编码的给定地址帐户中当前余额，以wei为单位。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  -d '{"jsonrpc":"2.0","method":"eth_getBalance","params":
["0xc94770007dda54cF92009BFF0dE90c06F603a09f", "latest"],"id":1}'
```


4.4.1.1.24 eth_subscribe

简介

为特定事件创建新订阅。节点返回订阅 ID。对于与订阅匹配的每个事件，将发送包含相关数据的通知以及订阅 ID。该API所消耗的计算单元为10。

参数说明

参数	类型	说明
事件类型	String	指定要侦听的事件类型。
可选参数	String	要包含的可选参数，用于描述要侦听的事件类型，包括newHeads、newPendingTransactions、logs。

返回值

当订阅处于活动状态时，您将收到格式化为如下对象的事件：

事件对象：

- jsonrpc：始终为“2.0”。
- method：始终“eth_subscription”。
- params：具有以下字段的对象：
  - subscription：创建此订阅的调用返回的订阅 ID。此 ID 将附加到所有收到的事件，也可用于使用eth_unsubscribe。
  - result：内容因事件类型而异的对象。

请求样式

```
wscat -c wss://your-http-endpoint/v1/<API-KEY> -x '{"jsonrpc":"2.0", "id": 1, "method": "eth_subscribe", "params": ["logs"]}'
```

4.4.1.1.25 eth_unsubscribe

简介

通过使用订阅 ID 调用此方法来取消订阅。它返回一个布尔值，指示订阅已成功取消。该API所消耗的计算单元为10。

参数说明

参数	类型	说明
订阅ID	String	要取消订阅的订阅的 ID。

返回值

如果订阅已成功取消，返回True，否则返回False。

请求样式

```
wscat -c wss://your-http-endpoint/v1/<API-KEY> -x '{"jsonrpc":"2.0", "id": 1, "method": "eth_unsubscribe",
"params": ["0x9cef478923ff08bf67fde6c64013158d"]}'
```

4.4.1.1.26 eth_getStorageAt

简介

返回给定地址的存储位置的值。该API所消耗的计算单元为18。

参数说明

参数	类型	说明
地址	String	表示存储地址（20 字节）的字符串。
存储位置	String	存储位置的十六进制代码。
区块编号	String	想要查询的区块的十六进制的编号，或者是字符串"earliest"、"latest"、"pending"。

返回值

所提供存储位置的值。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  -d '{"jsonrpc":"2.0","method":"eth_getStorageAt","params":
["0x295a70b2de5e3953354a6a8344e616ed314d7251",
"0x6661e9d6d8b923d5bbaab1b96e1dd51ff6ea2a93520fdc9eb75d059238b8c5e9", "0x65a8db"],"id":1}'
```

4.4.1.1.27 eth_accounts

简介

返回客户端拥有的地址列表。该API所消耗的计算单元为12。

参数说明

此方法不接受任何参数。

返回值

数组，包含客户端拥有的地址的十六进制编码的字符串。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
    -X POST \
    -H "Content-Type: application/json" \
    -d '{"jsonrpc":"2.0","method":"eth_accounts","params":[],"id":1}'
```

4.4.1.1.28 eth_getCode

简介

返回给定地址处智能合约的已编译字节代码（如果有）。该API所消耗的计算单元为19。

参数说明

参数	类型	说明
地址	String	表示存储地址（20 字节）的字符串，从中获取编译后的字节码。
区块编码	String	想要查询的区块的十六进制的编号，或者是字符串"earliest"、"latest"、"pending"。

返回值

给定地址处智能合约的已编译字节代码。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  -d '{"jsonrpc":"2.0","method":"eth_getCode","params":
["0x06012c8cf97bead5deae237070f9587f8e7a266d", "0x65a8db"],"id":1}'
```

4.4.1.1.29 eth_getProof

简介

返回指定账户的账户和存储值，包括 Merkle 证明。该API所消耗的计算单元为34。

参数说明

参数	类型	说明
账户地址	String	表示存储地址（20 字节）的字符串，从中获取编译后的字节码。
存储键	Array	要验证和包含的 32 字节存储键值的数组。
区块编号	String	想要查询的区块的十六进制的编号，或者是字符串 "earliest"、"latest"、"pending"。

返回值

- address：与账户相关的地址。
- accountProof：RLP 序列化 MerkleTree-Nodes 的数组，从 stateRoot-Node 开始，遵循 SHA3（地址）的路径作为键。
- balance：当前余额的十六进制，以 wei 为单位。
- codeHash：账户代码的 32 字节哈希值。
- nonce：账户的随机数。
- storageHash: 32 字节。StorageRoot 的 SHA3。所有存储都将从这里开始提供 Merkle 证明rootHash。
- storageProof：请求的存储条目数组。每个条目都是一个具有以下属性的对象：
  - key：请求的存储密钥。
  - value：存储值。
  - proof：RLP 序列化 MerkleTree-Nodes 的数组，从 storageHash-Node 开始，遵循 SHA3（密钥）的路径作为路径。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  -d '{"jsonrpc": "2.0","method": "eth_getProof","id": 1,"params":
["0x7F0d15C7FAae65896648C8273B6d7E43f58Fa842",
["0x56e81f171bcc55a6ff8345e692c0f86e5b48e01b996cadc001622fb5e363b421"], "latest"]}'
```

4.4.1.1.30 eth_getLogs

简介

返回与给定过滤器对象匹配的所有日志的数组。该API所消耗的计算单元为75。

参数说明

参数	类型	说明
address	String	可选参数，合约地址（20字节）或日志应源自的地址列表。
fromBlock	String	可选参数，默认为“latest”，十六进制区块号，或字符串latest,earliest或pending。
toBlock	String	可选参数，默认为“latest”，十六进制区块号，或字符串latest,earliest或pending。
topics	String	可选参数，32 字节 DATA 主题数组。主题与顺序相关。
blockhash	String	可选参数，将返回的日志限制为 32 字节哈希中引用的单个块blockHash。使用blockHash相当于设置fromBlock和 toBlock中引用的区块均为blockHash所对应的区块。如果blockHash出现在过滤条件中，则fromBlock和toBlock都不能够被设置。

返回值

日志对象数组，如果自上次轮询以来没有任何更改，则为空数组。日志对象包含以下键及其值：

- removed：若日志由于链重组而被删除，则返回true。如果它是有效的日志，则返回false。
- logIndex：块中日志索引位置的十六进制。当它是待处理（Pending）日志时返回NULL。
- transactionIndex：创建日志的事务索引位置的十六进制。当它是待处理（Pending）日志时返回NULL。
- transactionHash: 32 字节。创建此日志的事务的哈希值。当它是待处理（Pending）日志时返回NULL。
- blockHash: 32 字节。该日志所在块的哈希值，当它是待处理（Pending）日志时返回NULL。
- blockNumber：该日志所在的块号，当它是待处理（Pending）日志时返回NULL。

- address: 20 字节。该日志的来源地址。
- data：包含一个或多个 32 字节非索引日志参数。
- topics：包含 0 到 4 个索引日志参数的数组，每个 32 字节。在 Solidity 中，第一个主题是事件签名的哈希值（例如 Deposit(address,bytes32,uint256)），除非您使用匿名说明符声明事件。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  -d '{"jsonrpc":"2.0","method":"eth_getLogs","params":[{"blockHash":
"0x7c5a35e9cb3e8ae0e221ab470abae9d446c3a5626ce6689fc777dcffcab52c70", "topics":
["0x241ea03ca20251805084d27d4440371c34a0b85ff108f6bb5611248f73818b80"]}],"id":74}'
```

4.4.1.1.31 eth_getFilterChanges

简介

过滤器的轮询方法，返回自上次轮询以来发生的日志数组。过滤器必须通过调用eth_newFilter、eth_newBlockFilter、eth_newPendingTransactionFilter来创建。该API所消耗的计算单元为26。

参数说明

参数	类型	说明
过滤器ID	String	表示过滤器 ID 的字符串。

返回值

- log object array：（数组）日志对象数组，如果自上次轮询以来没有任何更改，则为空数组。
- 对于使用eth_newBlockFilter返回值创建的过滤器，返回值是块哈希（32 字节），例如["0x3454645634534..."]。
- 对于使用eth_newFilter日志创建的过滤器，对象具有以下参数：
  - address: 该日志的来源地址。
  - blockHash:该日志所在块的哈希值。当它是待处理（Pending）日志时返回NULL。
  - blockNumber：该日志所在的块号。当它是待处理（Pending）日志时返回NULL。
  - data： 包含日志的非索引参数。
  - logIndex：块中日志索引位置的十六进制。当它是待处理（Pending）日志时返回NULL。
  - removed：若日志由于链重组而被删除，则返回true。如果它是有效的日志，则返回false。
  - topics：数据数组。索引日志参数的 0 到 4 个 32 字节 DATA 的数组。在 Solidity 中，第一个topic是事件签名的哈希值（例如 Deposit(address,bytes32,uint256)），除非您使用匿名说明符声明事件。

- transactionHash: 32 字节。创建此日志的事务的哈希值。当它是待处理（Pending）日志时返回NULL。
- transactionIndex：创建此日志的事务索引位置的十六进制。当它是待处理（Pending）日志时返回NULL。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
    -X POST \
    -H "Content-Type: application/json" \
    -d '{"jsonrpc":"2.0","method":"eth_getFilterChanges","params":["0x16"],"id":73}'
```

4.4.1.1.32 eth_getFilterLogs

简介

返回与给定过滤器 ID 匹配的所有日志的数组。该API所消耗的计算单元为75。

参数说明

参数	类型	说明
address	String	可选参数，合约地址（20字节）或日志应源自的地址列表。
fromBlock	String	可选参数，默认为“latest”，十六进制区块号，或字符串latest，earliest或pending。
toBlock	String	可选参数，默认为“latest”，十六进制区块号，或字符串latest，earliest或pending。
topics	String	可选参数，32 字节 DATA 主题数组。主题与顺序相关。
blockhash	String	可选参数，将返回的日志限制为 32 字节哈希中引用的单个块blockHash。使用blockHash相当于设置fromBlock和 toBlock中引用的区块均为blockHash所对应的区块。如果blockHash出现在过滤条件中，则fromBlock和toBlock都不能够被设置。

返回值

- log 对象数组：与过滤器匹配的日志对象数组。对于自上次轮询以来发生的日志数组，请使用eth_getFilterChanges。日志对象包含以下键及其值：
  - address: 该日志的来源地址。
  - blockHash：该日志所在块的哈希值。当它是待处理（Pending）日志时返回NULL。
  - blockNumber：该日志所在的块号。当它是待处理（Pending）日志时返回NULL。
  - data：包含日志的非索引参数。
  - logIndex：块中日志索引位置的十六进制。当它是待处理（Pending）日志时返回NULL。

- removed：若日志由于链重组而被删除，则返回true。如果它是有效的日志，则返回false。
- topics：数据数组。索引日志参数的 0 到 4 个 32 字节 DATA 的数组。在 Solidity 中，第一个topic是事件签名的哈希值（例如 Deposit(address,bytes32,uint256)），除非您使用匿名说明符声明事件。
- transactionHash: 创建此日志的事务的哈希值。当它是待处理（Pending）日志时返回NULL。
- transactionIndex：创建此日志的事务索引位置的十六进制。当它是待处理（Pending）日志时返回NULL。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
   -X POST \
   -H "Content-Type: application/json" \
   -d '{"jsonrpc":"2.0","method":"eth_getFilterLogs","params":["0x16"],"id":74}'
```

4.4.1.1.33 eth_newBlockFilter

简介

在节点中创建一个过滤器，以在新区块到达时发出通知。该API所消耗的计算单元为20。

参数说明

此方法不接受任何参数。

返回值

十六进制表示的新创建的过滤器ID。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
   -X POST \
   -H "Content-Type: application/json" \
   -d '{"jsonrpc":"2.0","method":"eth_newBlockFilter","params":[],"id":73}'
```

4.4.1.1.34 eth_newFilter

简介

根据给定的过滤器选项创建过滤器对象，以在状态更改（日志）时发出通知。该API所消耗的计算单元为20。

参数说明

参数	类型	说明
address	String	可选参数，合约地址（20字节）或日志应源自的地址列表。

参数	类型	说明
fromBlock	String	可选参数，默认为“latest”，十六进制区块号，或字符串latest，earliest或pending。
toBlock	String	可选参数，默认为“latest”，十六进制区块号，或字符串latest，earliest或pending。
topics	String	可选参数，32 字节 DATA 主题数组。主题与顺序相关。

返回值

十六进制表示的新创建的过滤器ID。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
    -X POST \
    -H "Content-Type: application/json" \
    -d '{"jsonrpc":"2.0","method":"eth_newFilter","params":[{"topics":
["0xddf252ad1be2c89b69c2b068fc378daa952ba7f163c4a11628f55a4df523b3ef"]}],"id":73}'
```

4.4.1.1.35 eth_newPendingTransactionFilter

简介

在节点中创建一个过滤器，以在新的待处理事务到达 Arbitrum 时发出通知。该API所消耗的计算单元为20。

参数说明

此方法不接受任何参数。

返回值

十六进制表示的新创建的过滤器ID。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
    -X POST \
    -H "Content-Type: application/json" \
    -d '{"jsonrpc":"2.0","method":"eth_newPendingTransactionFilter","params":[],"id":73}'
```

4.4.1.1.36 eth_uninstallFilter

简介

卸载具有给定 id 的过滤器。当不再需要观察时调用。此外，如果一段时间内没有通过eth_getFilterChanges请求过滤器，过滤器就会超时。该API所消耗的计算单元为12。

参数说明

参数	类型	说明
过滤器ID	String	要卸载的过滤器ID的字符串。

返回值

如果过滤器已成功卸载，返回true，否则false。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
   -X POST \
   -H "Content-Type: application/json" \
   -d '{"jsonrpc":"2.0","method":"eth_uninstallFilter","params":["0xb"],"id":73}'
```

4.4.1.1.37 eth_chainId

简介

返回当前配置的链 ID。该API所消耗的计算单元为1。

参数说明

此方法不接受任何参数。

返回值

十六进制表示的链 ID。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
   -X POST \
   -H "Content-Type: application/json" \
   -d '{"jsonrpc":"2.0","method":"eth_chainId","params": [],"id":1}'
```

4.4.1.1.38 web3_sha3

简介

返回给定数据的 Keccak-256 编码结果（不是标准化 SHA3-256 ）。该API所消耗的计算单元为15。

参数说明

参数	类型	说明
数据	String	需要转换的数据

返回值

给定输入的SHA3编码后的结果。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
   -X POST \
   -H "Content-Type: application/json" \
   -d '{"jsonrpc":"2.0","method":"web3_sha3","params": ["0x68656c6c6f20776f726c64"],"id":64}'
```

4.4.1.1.39 web3_clientVersion

简介

返回当前客户端的版本。该API所消耗的计算单元为15。

参数说明

此方法不接受任何参数。

返回值

客户端的版本。

5 BNB Smart Chain

5.1 BNB Smart Chain 介绍

BNB Smart Chain（BSC）是一个与币安链并行的区块链，拥有智能合约功能并与以太坊虚拟机（EVM）兼容。该链依赖于一个由55个验证器组成的系统，该系统具有权益证明（PoSA）共识，可以支持短阻塞时间和更低的费用。BSC链的目标是保持完整币安链的高吞吐量，同时将智能合约引入其生态系统。

BNB Smart Chain官方链接：BNB Smart Chain介绍

用户可以通过使用华为云公链节点引擎，来提升区块链使用与开发的效率，增强其稳定性与私密性。**华为云将永远不会收集用户的区块链地址。**

说明

- **支持网络**
  - BNB Smart Chain：HTTP，WebSocket
- **BNB Smart Chain API列表**

5.2 HTTP 请求示例

5.2.1 使用 cURL 发送 HTTP API 请求

Request example:

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"eth_getBlockByNumber","params":["0xf8e7d",false],"id":1,"jsonrpc":"2.0"}'
```

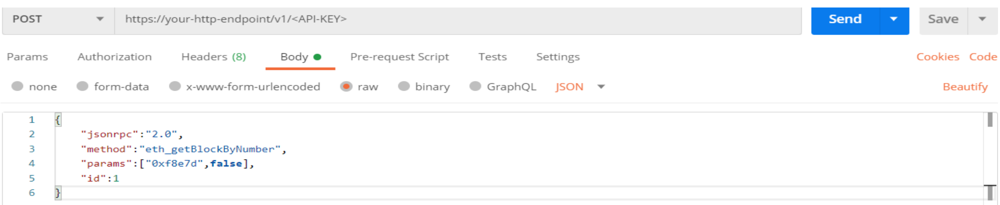
Response example:

```
{
    "jsonrpc": "2.0",
    "id": 1,
    "result": {
        "baseFeePerGas": "0x0",
        "difficulty": "0x2",
```

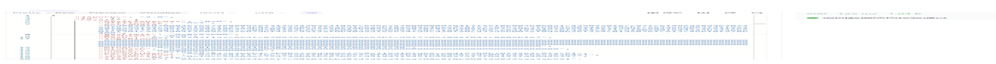
```
        "extraData":
"0xd883010300846765746888676f312e32302e31856c696e7578000000394f7f55f8b003b860a1c43902a0f6cc97fc61995d28ddb7074edf5c69fe4c503b5194c152cf3b827de2e8b32f6a5d1e6aab282e84e46ee978157c85676c3fb964e90a42ef49ebfc564ae6b863f41f95228784da1315d5f1ba50749b39dbdc0212db7e7920519b985ef84a830f8e7ba05fbae7c490271975e55112f31add5c69a35a46c2b53f1629f18b477f9a8ef6ee830f8e7ca08384763b6b4cab00a7a9dd0de27e8e319d10fd66ce0d16708b70f4264c8f2c6180e801644243b324bb502cf301ca933d84f26895aa7607dbf425c9a12af0b48dad1ee634fddcd28fa0ca33d6b007ef8da8b26f224c81d5876fce13c46bf8fa78b400",
        "gasLimit": "0x8583b00",
        "gasUsed": "0x0",
        "hash": "0x7f222d3f1a7c664fc8709361e0f0d0e60ae63bfcd9f770e7892f043c8885b167",
        "logsBloom":
"0x00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000",
        "miner": "0x69cb38199d2c2419b384fa6e22f4667069843730",
        "mixHash":
"0x0000000000000000000000000000000000000000000000000000000000000000",
        "nonce": "0x0000000000000000",
        "number": "0xf8e7d",
        "parentHash":
"0x8384763b6b4cab00a7a9dd0de27e8e319d10fd66ce0d16708b70f4264c8f2c61",
        "receiptsRoot":
"0x56e81f171bcc55a6ff8345e692c0f86e5b48e01b996cadc001622fb5e363b421",
        "sha3Uncles":
"0x1dcc4de8dec75d7aab85b567b6ccd41ad312451b948a7413f0a142fd40d49347",
        "size": "0x316",
        "stateRoot":
"0xbb246175024b46f7c985032bba83d3d5aa42ea708e8f8b90834d8445aa3ab651",
        "timestamp": "0x655d6eca",
        "totalDifficulty": "0x1f1b52",
        "transactions": [],
        "transactionsRoot":
"0x56e81f171bcc55a6ff8345e692c0f86e5b48e01b996cadc001622fb5e363b421",
        "uncles": []
    }
}
```


5.2.2 使用 post-man 发送 HTTP API 请求

Request example:



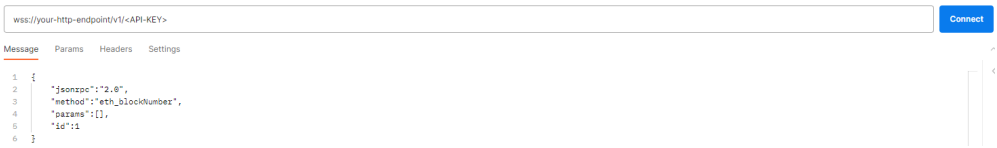
Response example:



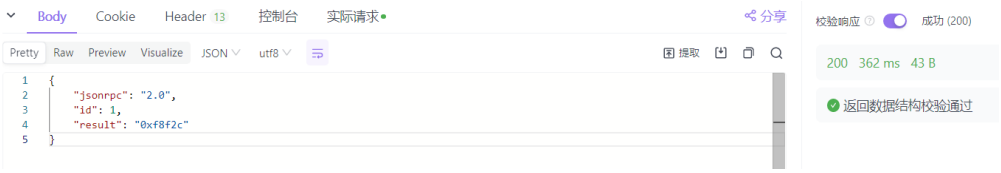
5.3 WebSocket 请求示例

5.3.1 使用 post-man 发送 JSON-RPC API 请求

Request example:



Response example:



5.4 BNB Smart Chain API 列表

5.4.1 专享版

表 5-1 可用 BNB Smart Chain API 列表

API方法	说明	流控值（次/s）
debug_traceBlockByNumber	追踪指定区块号中的所有交易的详细调用信息，包括调用类型、发送者、接收者地址、交易值、gas、输入数据、输出数据等。	5
debug_traceBlockByHash	追踪指定区块哈希中的所有交易的详细调用信息，包括调用类型、发送者、接收者地址、交易值、gas、输入数据、输出数据等。	15
debug_traceBlock	用于返回一个区块中包含的所有交易的完整操作码调用栈追踪。	50
debug_traceTransaction	追踪指定的交易。	200
eth_accounts	返回客户端拥有的地址列表。	30000
eth_call	立即执行新的消息调用，而不在区块链上创建交易。	24500

API方法	说明	流控值（次/s）
eth_estimateGas	返回给定交易的所消耗的Gas的估计值。	2500
eth_getFilterLogs	返回与给定过滤器 ID 匹配的所有日志的数组。	1900
eth_getLogs	返回与给定过滤器对象匹配的所有日志的数组。	10
eth_getFilterChanges	过滤器的轮询方法，返回自上次轮询以来发生的日志数组。过滤器必须通过调用eth_newFilter、eth_newBlockFilter、eth_newPendingTransactionFilter来创建。	39000
eth_blockNumber	返回区块链的最新区块号。	40000
debug_traceCall	通过在给定块执行的上下文中执行eth调用来返回可能的跟踪结果。	1000
eth_chainId	返回当前配置的链 ID。	34000
eth_feeHistory	返回历史消耗的Gas信息的集合。	30000
eth_gasPrice	返回当前的Gas价格（以 wei 为单位）。	41000
eth_getBalance	返回给定地址的帐户余额。	27000
eth_getBlockByHash	返回与给定区块哈希匹配的区块的信息。	• 返回完整的区块对象：2000 • 不返回完整区块对象：22000
eth_getBlockByNumber	返回与给定的区块号匹配的区块信息。	• 返回完整的区块对象：2000 • 不返回完整区块对象：22000
eth_getBlockTransactionCountByHash	返回与给定块哈希匹配的区块的交易数。	29000
eth_getBlockTransactionCountByNumber	返回与给定区块编号匹配的区块的交易数。	32000
eth_getCode	返回给定地址处智能合约的已编译字节代码（如果有）。	27000

API方法	说明	流控值（次/s）
eth_getProof	返回指定账户的账户和存储值，包括 Merkle 证明。	11000
eth_getStorageAt	返回给定地址的存储位置的值。	24000
eth_getTransactionByBlockHashAndIndex	返回给定交易哈希和交易索引位置的交易信息。	28000
eth_getTransactionByBlockNumberAndIndex	返回给定区块号和交易索引位置的交易信息。	30000
eth_getTransactionByHash	根据交易哈希返回有关交易的信息。	25000
eth_getTransactionCount	返回从某一地址发送的交易数。	28000
eth_getTransactionReceipt	通过交易哈希返回交易的收据。	21000
eth_getUncleByBlockHashAndIndex	返回给定区块哈希和索引位置的叔区块信息。	6000
eth_getUncleByBlockNumberAndIndex	返回给定区块号和索引位置的叔区块信息。	6000
eth_getUncleCountByBlockHash	返回与给定区块哈希匹配的区块中叔区块的数量。	30000
eth_getUncleCountByBlockNumber	返回与给定区块编号匹配的区块中叔区块的数量。	32000
eth_maxPriorityFeePerGas	返回每个Gas的费用，这是您可以支付多少优先费用或“小费”的估计，以获得当前区块中包含的交易。	38000
eth_newBlockFilter	在节点中创建一个过滤器，以在新区块到达时发出通知。	29000
eth_newFilter	根据给定的过滤器选项创建过滤器对象，以在状态更改（日志）时发出通知。	24000
eth_newPendingTransactionFilter	在节点中创建一个过滤器，以在新的待处理事务到达BNB Smart Chain时发出通知。	80

API方法	说明	流控值（次/s）
eth_sendRawTransaction	创建新的消息调用交易或为签名交易创建合约。	2500
eth_subscribe	为特定事件创建新订阅。节点返回订阅 ID。对于与订阅匹配的每个事件，将发送包含相关数据的通知以及订阅 ID。	1000
eth_syncing	返回当前的同步状态。	35000
eth_uninstallFilter	卸载具有给定 id 的过滤器。当不再需要观察时调用。此外，如果一段时间内没有通过eth_getFilterChanges请求过滤器，过滤器就会超时。	33000
eth_unsubscribe	通过使用订阅 ID 调用此方法来取消订阅。它返回一个布尔值，指示订阅已成功取消。	1000
net_listening	当客户端正在主动侦听网络连接时为true。	32000
net_version	返回当前网络id。	34000
txpool_content	返回所有挂起和排队的交易。	40
txpool_inspect	返回所有挂起和排队的交易的文本摘要。	200
txpool_status	返回处于待处理（Pending）状态和排队（Queued）状态的事务数。	25000
web3_clientVersion	返回当前客户端的版本。	30000
web3_sha3	返回给定数据的Keccak-256 编码结果（不是标准化 SHA3-256）。	32000
eth_hashrate	返回节点每秒计算的哈希值数量。	40000
eth_mining	如果节点正在主动挖掘新块，则返回true。	42000
nes_sendGasOptimizedTransaction	返回用于查询该Gas优化交易状态的id。	250

API方法	说明	流控值（次/s）
nes_getGasOptimizedTransactionStatus	返回增强交易的状态。	1500

5.4.2 共享版

5.4.2.1 eth_blocknumber

简介

返回区块链的最新区块号。该API所消耗的计算单元为105。

参数说明

此方法不接受任何参数。

返回值

十六进制编码的最新区块号。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"eth_blockNumber","params":[],"id":1,"jsonrpc":"2.0"}'
```

5.4.2.2 eth_getBlockByNumber

简介

返回与给定的区块号匹配的区块信息。该API所消耗的计算单元为133。

参数说明

参数	类型	说明
区块编号	String	十六进制的区块编号，或者是字符串 "earliest"、"latest"、"pending"。
交易详细信息标志	Bool	当此值为 true 时，该方法返回完整的交易对象，否则，它仅返回交易的哈希。

返回值

- Object - 区块对象，如果未找到区块，则为 null。区块对象包含以下字段：
  - number: 编码为十六进制的请求块的块号。如果是一个处于待处理（Pending）状态的区块，则为空。
  - hash: 区块的哈希。如果是一个处于待处理（Pending）状态的区块，则为空。
  - parentHash: 父区块的哈希。
  - nonce: 生成的工作量证明的哈希值。如果是一个处于待处理（Pending）状态的区块，则为空。
  - sha3Uncles: 区块中叔区块数据的 SHA3。
  - logsBloom: 区块日志的布隆过滤器。如果是一个处于待处理（Pending）状态的区块，则为空。
  - transactionsRoot: 区块中交易树的根。
  - stateRoot： 区块的最终状态树的根。
  - receiptsRoot： 区块的收据树的根。
  - miner： 获得采矿奖励的受益人的地址。
  - difficulty：此区块的难度。
  - totalDifficulty：直到这个区块时，链的总难度。
  - extraData：此区块的“额外数据”字段。
  - size：此区块的大小（以字节为单位）。
  - gasLimit：此区块中允许的最大gas。
  - gasUsed：此区块中所有交易的总使用gas。
  - timestamp：整理区块时的 unix 时间戳。
  - transactions：交易对象的数组，或 32 字节的交易哈希，具体取决于最后一个给定的参数。
  - uncles：叔区块哈希数组。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"eth_getBlockByNumber","params":["0xc5043f",false],"id":1,"jsonrpc":"2.0"}'
```

5.4.2.3 eth_hashrate

简介

返回节点每秒计算的哈希值数量。该API所消耗的计算单元为104。

参数说明

此方法不接受任何参数。

返回值

十六进制编码的每秒哈希数。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"eth_hashrate","params":[],"id":1,"jsonrpc":"2.0"}'
```

5.4.2.4 eth_getUncleCountByBlockNumber

简介

返回与给定区块编号匹配的区块中叔区块的数量。该API所消耗的计算单元为130。

参数说明

参数	类型	说明
区块编号	String	想要查询的区块的十六进制的编号。

返回值

区块中叔区块的数量，以十六进制为编码。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"eth_getUncleCountByBlockNumber","params":["0xc5043f"],"id":1,"jsonrpc":"2.0"}'
```

5.4.2.5 eth_getUncleCountByBlockHash

简介

返回与给定区块哈希匹配的区块中叔区块的数量。该API所消耗的计算单元为136。

参数说明

参数	类型	说明
区块哈希	String	想要查询的区块的哈希值。

返回值

区块中叔区块的数量，以十六进制为编码。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
```

```
 --data '{"method":"eth_getUncleCountByBlockHash","params":
["0x81e807e7a6031d9f103eeee2a2edc5994c3432ee1e3227c66ff78eef30ea1dec"],"id":1,"jsonrpc":"2.0"}'
```

5.4.2.6 eth_getBlockByHash

简介

返回与给定区块哈希匹配的区块的信息。该API所消耗的计算单元为145。

参数说明

参数	类型	说明
区块哈希	String	想要查询的区块的哈希值。
事务详细信息标志	Bool	当此值为 true 时，该方法返回完整的交易对象，否则，它仅返回交易的哈希。

返回值

Object - 区块对象，如果未找到区块，则为 null。区块对象包含以下字段：

- number: 编码为十六进制的请求块的块号。如果是一个处于待处理（Pending）状态的区块，则为空。
- hash: 区块的哈希。如果是一个处于待处理（Pending）状态的区块，则为空。
- parentHash: 父区块的哈希。
- nonce: 生成的工作量证明的哈希值。如果是一个处于待处理（Pending）状态的区块，则为空。
- sha3Uncles: 区块中叔区块数据的 SHA3。
- logsBloom: 区块日志的布隆过滤器。如果是一个处于待处理（Pending）状态的区块，则为空。
- transactionsRoot: 区块中交易树的根。
- stateRoot： 区块的最终状态树的根。
- receiptsRoot： 区块的收据树的根。
- miner： 获得采矿奖励的受益人的地址。
- difficulty：此区块的难度。
- totalDifficulty： 直到这个区块时，链的总难度。
- extraData：此区块的“额外数据”字段。
- size：此区块的大小（以字节为单位）。
- gasLimit：此区块中允许的最大gas。
- gasUsed：此区块中所有交易的总使用gas。
- timestamp：整理区块时的 unix 时间戳。
- transactions：交易对象的数组，或 32 字节的交易哈希，具体取决于最后一个给定的参数。

- uncles：叔区块哈希数组。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"eth_getBlockByHash","params":
["0x81e807e7a6031d9f103eeee2a2edc5994c3432ee1e3227c66ff78eef30ea1dec",false],"id":1,"jsonrpc":"2.0"}'
```

5.4.2.7 eth_getTransactionByHash

简介

根据交易哈希返回有关交易的信息。该API所消耗的计算单元为120。

参数说明

参数	类型	说明
交易哈希	String	要查询的交易的哈希值。

返回值

Object - 交易对象，如果未找到交易，则为 null。交易对象包含以下字段：

- blockHash：此交易所在的区块的哈希值。当它是待处理的（Pending）日志时为 null
- blockNumber：此交易所在的区块号。当它是待处理的（Pending）日志时为 null
- from：发件人的地址
- gas：发送方提供的gas，编码为十六进制
- gasPrice：发件人提供的 wei 格式的gas价格，编码为十六进制
- maxFeePerGas：交易中设置的每种gas的最高值
- maxPriorityFeePerGas：交易中设置的最高优先级gas
- hash：交易的哈希值
- input：与交易一起发送的数据
- nonce：发送方在此交易之前进行的交易数，编码为十六进制
- to：接收方的地址。当它是合约创建交易时为 null
- transactionIndex：从中创建日志的交易索引位置的整数。当它是待处理的（Pending）日志时为 null
- value：以 wei 为单位的十六进制编码的转账数额
- type：交易类型
- accessList：交易计划访问的地址和存储密钥的列表
- chainId：交易的链 ID（如果有）
- v：签名的标准化 V 字段

- r：签名的 R 字段
- s：签名的 S 字段

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"eth_getTransactionByHash","params":
["0xb142342a7fd70602b7a0ba3688a41bfcbb4fbc3490c252ca48af2594619d220c"],"id":1,"jsonrpc":"2.0"}'
```

5.4.2.8 eth_getTransactionCount

简介

返回从某一地址发送的交易数。该API所消耗的计算单元为148。

参数说明

参数	类型	说明
地址	String	要检查的交易计数的地址。
区块编号	String	想要查询的区块的十六进制的编号，或者是字符串"earliest"、"latest"、"pending"。

返回值

从地址发送的十六进制编码的交易数量

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"eth_getTransactionCount","params":
["0x8D97689C9818892B700e27F316cc3E41e17fBeb9", "latest"],"id":1,"jsonrpc":"2.0"}'
```

5.4.2.9 eth_getTransactionByBlockHashAndIndex

简介

返回给定交易哈希和交易索引位置的交易信息。该API所消耗的计算单元为149。

参数说明

参数	类型	说明
交易哈希	String	想要查询的交易的哈希值。

参数	类型	说明
索引	String	编码为十六进制的交易索引位置。

返回值

Object - 交易对象，如果未找到交易，则为 null。交易对象包含以下字段：

- blockHash：此交易所在的区块的哈希值。当它是待处理的（Pending）日志时为 null
- blockNumber：此交易所在的区块号。当它是待处理的（Pending）日志时为 null
- from：发件人的地址
- gas：发送方提供的gas，编码为十六进制
- gasPrice：发件人提供的以 wei 为单位的gas价格，编码为十六进制
- maxFeePerGas：交易中设置的每种gas的最高值
- maxPriorityFeePerGas：交易中设置的最高优先级gas
- hash：交易的哈希值
- input：与交易一起发送的数据
- nonce：发送方在此交易之前进行的交易数，编码为十六进制
- to：接收方的地址。当它是合约创建交易时为 null
- transactionIndex：从中创建日志的交易索引位置的整数。当它是待处理的（Pending）日志时为 null
- value：以 wei 为单位的十六进制编码的转账数额
- type：交易类型
- accessList：交易计划访问的地址和存储密钥的列表
- chainId：交易的链 ID（如果有）
- v：签名的标准化 V 字段
- r：签名的 R 字段
- s：签名的 S 字段

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"eth_getTransactionByBlockHashAndIndex","params":
["0x81e807e7a6031d9f103eeee2a2edc5994c3432ee1e3227c66ff78eef30ea1dec","0x0"],"id":1,"jsonrpc":"2.0"}
'
```

5.4.2.10 eth_getTransactionByBlockNumberAndIndex

简介

返回给定区块号和交易索引位置的交易信息。该API所消耗的计算单元为137。

参数说明

参数	类型	说明
区块编号	String	想要查询的区块的十六进制的编号，或者是字符串"earliest"、"latest"、"pending"。
索引	String	编码为十六进制的交易索引位置。

返回值

Object - 交易对象，如果未找到交易，则为 null。交易对象包含以下字段：

- blockHash：此交易所在的区块的哈希值。当它是待处理的（Pending）日志时为 null
- blockNumber：此交易所在的区块号。当它是待处理的（Pending）日志时为 null
- from：发件人的地址
- gas：发送方提供的gas，编码为十六进制
- gasPrice：发件人提供的以 wei 为单位的gas价格，编码为十六进制
- maxFeePerGas：交易中设置的每种gas的最高值
- maxPriorityFeePerGas：交易中设置的最高优先级gas
- hash：交易的哈希值
- input：与交易一起发送的数据
- nonce：发送方在此交易之前进行的交易数，编码为十六进制
- to：接收方的地址。当它是合约创建交易时为 null
- transactionIndex：从中创建日志的交易索引位置的整数。当它是待处理的（Pending）日志时为 null
- value：以 wei 为单位的十六进制编码的转账数额
- type：交易类型
- accessList：交易计划访问的地址和存储密钥的列表
- chainId：交易的链 ID（如果有）
- v：签名的标准化 V 字段
- r：签名的 R 字段
- s：签名的 S 字段

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"eth_getTransactionByBlockNumberAndIndex","params":["0xc5043f",
"0x0"],"id":1,"jsonrpc":"2.0"}'
```

5.4.2.11 eth_getBlockTransactionCountByHash

简介

返回与给定块哈希匹配的区块的交易数。该API所消耗的计算单元为143。

参数说明

参数	类型	说明
区块哈希	String	想要查询的区块的哈希值。

返回值

以十六进制格式表示的所查询区块中的交易数。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"eth_getBlockTransactionCountByHash","params":
["0x81e807e7a6031d9f103eeee2a2edc5994c3432ee1e3227c66ff78eef30ea1dec"],"id":1,"jsonrpc":"2.0"}'
```

5.4.2.12 eth_getBlockTransactionCountByNumber

简介

返回与给定区块编号匹配的区块的交易数。该API所消耗的计算单元为128。

参数说明

参数	类型	说明
区块编号	String	想要查询的区块的十六进制的编号，或者是字符串"earliest"、"latest"、"pending"。

返回值

以十六进制格式表示的所查询区块中的交易数。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"eth_getBlockTransactionCountByNumber","params":["0xc5043f"],"id":1,"jsonrpc":"2.0"}'
```

5.4.2.13 eth_syncing

简介

返回当前的同步状态。该API所消耗的计算单元为118

参数说明

此方法不接受任何参数。

返回值

交易收据对象的数组，其中每个交易对象包含如下内容：

返回值1：

Boolean - 若同步完成，返回false

返回值2：

Object - 若同步中，返回同步数据状态

- startingBlock：导入的开始区块号，编码为十六进制
- currentBlock：当前的区块号，与**eth_blockNumber**结果相同，编码为十六进制
- highestBlock：预估的最高区块号，编码为十六进制

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
   -X POST \
   -H "Content-Type: application/json" \
   -d '{"jsonrpc":"2.0","method":"eth_syncing","params": [],"id":1}'
```

5.4.2.14 eth_getTransactionReceipt

简介

通过交易哈希返回交易的收据。该API所消耗的计算单元为120。

参数说明

参数	类型	说明
交易哈希	String	想要查询的交易的哈希值。

返回值

Object - 交易收据对象，如果未找到交易收据，则为 null。交易收据对象包含以下字段：

- blockHash：此交易所在的区块的哈希值
- blockNumber：添加此交易的区块号，编码为十六进制

- contractAddress：为创建合约创建的合约地址，如果并非合约创建则为空
- cumulativeGasUsed：在区块中执行此交易时使用的总gas
- effectiveGasPrice：为每单位gas支付的总基本费用加上额外交易费
- from：源地址
- gasUsed：仅此特定交易使用的gas
- logs：生成此交易的日志对象数组
  - address：生成此日志的地址
  - topics：索引日志参数的 0 到 4 个 32 字节 DATA 的数组。在 Solidity 中，第一个主题是事件签名的哈希值（例如 Deposit(address, bytes32, uint256)），除非您使用匿名说明符声明事件
  - data：日志的 32 字节非索引参数
  - blockNumber：此日志所在的块号
  - transactionHash：从中创建此日志的交易的哈希。如果日志处于待处理（Pending）状态，则为 null
  - transactionIndex：从中创建此日志的交易索引位置。如果日志处于待处理（Pending）状态，则为 null
  - blockHash：此日志所在的块的哈希值
  - logIndex：编码为十六进制的块中对数索引位置的整数。如果日志处于待处理（Pending）状态，则为 null
  - removed：如果日志由于链重组而被删除，则为 true，如果它是有效的日志，则为 false。
- logsBloom：用于检索相关日志的布隆过滤器
- status：1（成功）或 0（失败），编码为十六进制
- to：接收方的地址。当它是合约创建交易时为 null
- transactionHash：交易的哈希值
- transactionIndex：编码为十六进制的块中的交易索引位置
- type：值的类型

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"eth_getTransactionReceipt","params":
["0x6d755989f51032147484162c4dc3d6550552dbd8d3b094fe3c221bfa3c5942b2"],"id":1,"jsonrpc":"2.0"}'
```

5.4.2.15 eth_sendRawTransaction

简介

创建新的消息调用交易或为签名交易创建合约。该API所消耗的计算单元为120。

参数说明

参数	类型	说明
签名后的交易数据	String	使用您的私钥签名的交易。

返回值

交易哈希值，如果交易尚不可用，则为零哈希值。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"jsonrpc":"2.0","method":"eth_sendRawTransaction","params":["signed transaction"],"id":1}'
```

5.4.2.16 eth_call

简介

立即执行新的消息调用，而不在区块链上创建交易。该API所消耗的计算单元为120。

参数说明

包含交易的相关字段以及区块编号两部分。

参数	类型	说明
from	String	可选参数，发送交易的地址。
to	String	交易发送到的地址。
gas	Integer	可选参数，为交易执行提供的gas的整数。
gasPrice	Integer	可选参数，用于每个付费gas的gasPrice整数，编码为十六进制。
value	Integer	可选参数，与此交易一起发送的代币的数值，编码为十六进制。
data	String	可选参数，方法签名和编码参数的哈希值。有关更多信息，请参阅 Solidity 文档中的合约 ABI 描述。
区块编号	String	想要查询的区块的十六进制的编号，或者是字符串"earliest"、"latest"、"pending"。

返回值

执行合约方法的返回值。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"eth_call","params":
[{"from":null,"to":"0x6b175474e89094c44da98b954eedeac495271d0f","data":"0x70a08231000000000000000000000000006E0d01A76C3Cf4288372a29124A26D4353EE51BE"}, "latest"],"id":1,"jsonrpc":"2.0"}'
```

5.4.2.17 eth_mining

简介

如果节点正在主动挖掘新块，则返回true。该API所消耗的计算单元为99。

参数说明

此方法不接受任何参数。

返回值

如果节点正在主动挖掘新块，则返回true，否则返回false。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
   -X POST \
   -H "Content-Type: application/json" \
   -d '{"method":"eth_mining","params":[],"id":1,"jsonrpc":"2.0"}'
```

5.4.2.18 eth_estimateGas

简介

返回给定交易的所消耗的Gas的估计值。该API所消耗的计算单元为120。

参数说明

与 eth_call 的参数一致，但所有属性都是可选的。 如果没有指定Gas限制，geth 将使用来自待处理区块的区块Gas限制作为上限。 因此，当所需Gas数量高于待处理区块的Gas限制时，返回的估算值可能不足以执行调用/交易。

参数	类型	说明
from	String	发送交易的地址。
to	String	交易发送到的地址。
gas	Integer	为交易执行提供的gas的整数。
gasPrice	Integer	用于每个付费gas的gasPrice整数，编码为十六进制。
value	Integer	与此交易一起发送的代币的数值，编码为十六进制。

参数	类型	说明
data	String	方法签名和编码参数的哈希值。有关更多信息，请参阅 Solidity 文档中的合约 ABI 描述。
区块编号	String	想要查询的区块的十六进制的编号，或者是字符串 "earliest"、"latest"、"pending"。

返回值

交易所消耗Gas的预计数量。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"eth_estimateGas","params":
[{"from":"0x8D97689C9818892B700e27F316cc3E41e17fBeb9","to":"0xd3CdA913deB6f67967B99D67aCDFa1712C293601","value":"0x186a0"}],"id":1,"jsonrpc":"2.0"}'
```

5.4.2.19 eth_feeHistory

简介

返回历史消耗的Gas信息的集合。该API所消耗的计算单元为120。

参数说明

参数	类型	说明
区块数量	String/Integer	请求范围内的块数。在单个查询中可以请求 1 到 1024 个块。如果不是所有块都可用，它将返回小于请求的范围。
最新区块编号	String	想要查询的区块的十六进制的编号，或者是字符串 "earliest"、"latest"、"pending"。
奖励百分位数	Integer	可选参数，单调递增的百分位值列表，从每个区块的每种 gas 的有效优先费中采样，按升序排列，并按所使用的 gas 进行加权。

返回值

- oldestBlock：以十六进制数表示的返回范围中最早的区块编号。
- baseFeePerGas：数组，内容为每个 gas 的一系列区块基本费用，包括额外的区块值。额外的值是返回范围中最新块之后的下一个块。对于EIP-1559之前创建的块返回零。
- gasUsedRatio：数组，内容为每个区块gas使用比率。计算方式为gasUsed和gasLimit的比率。
- reward：来自单个区块的每个Gas数据点的有效优先费数组。如果块为空，则返回所有零。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  -d '{"id": 1, "jsonrpc": "2.0", "method": "eth_feeHistory", "params": ["0x5", "latest", [20,30]] }'
```

5.4.2.20 eth_maxPriorityFeePerGas

简介

返回每个Gas的费用，这是您可以支付多少优先费用或“小费”的估计，以获得当前区块中包含的交易。该API所消耗的计算单元为109。

参数说明

此方法不接受任何参数。

返回值

十六进制代码编码的当前区块中包含交易的每项 gas 费用。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
    -X POST \
    -H "Content-Type: application/json" \
    -d '{"jsonrpc":"2.0","method":"eth_maxPriorityFeePerGas","id":1}'
```

5.4.2.21 eth_gasPrice

简介

返回当前的Gas价格（以 wei 为单位）。该API所消耗的计算单元为101。

参数说明

此方法不接受任何参数。

返回值

以十六进制表示的当前Gas的价格，以 wei 为单位。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  -d '{"jsonrpc":"2.0","method":"eth_gasPrice","params": [],"id":1}'
```

5.4.2.22 eth_getBalance

简介

返回给定地址的帐户余额。该API所消耗的计算单元为120。

参数说明

参数	类型	说明
地址	String	表示用于检查余额的地址
区块编号	String	想要查询的区块的十六进制的编号，或者是字符串"earliest"、"latest"、"pending"。

返回值

以十六进制编码的给定地址帐户中当前余额，以wei为单位。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  -d '{"jsonrpc":"2.0","method":"eth_getBalance","params":
["0xc94770007dda54cF92009BFF0dE90c06F603a09f", "latest"],"id":1}'
```

5.4.2.23 eth_subscribe

简介

为特定事件创建新订阅。节点返回订阅 ID。对于与订阅匹配的每个事件，将发送包含相关数据的通知以及订阅 ID。该API所消耗的计算单元为120。

参数说明

参数	类型	说明
事件类型	String	指定要侦听的事件类型。

参数	类型	说明
可选参数	String	要包含的可选参数，用于描述要侦听的事件类型，包括newHeads、newPendingTransactions、logs。

返回值

当订阅处于活动状态时，您将收到格式化为如下对象的事件：

事件对象：

- jsonrpc：始终为“2.0”。
- method：始终“eth_subscription”。
- params：具有以下字段的对象：
  - subscription：创建此订阅的调用返回的订阅 ID。此 ID 将附加到所有收到的事件，也可用于使用eth_unsubscribe。
  - result：内容因事件类型而异的对象。

请求样式

```
wscat -c wss://your-http-endpoint/v1/<API-KEY> -x '{"jsonrpc":"2.0", "id": 1, "method": "eth_subscribe", "params": ["logs"]}'
```

5.4.2.24 eth_unsubscribe

简介

通过使用订阅 ID 调用此方法来取消订阅。它返回一个布尔值，指示订阅已成功取消。该API所消耗的计算单元为120。

参数说明

参数	类型	说明
订阅ID	String	要取消订阅的订阅的 ID。

返回值

如果订阅已成功取消，返回True，否则返回False。

请求样式

```
wscat -c wss://your-http-endpoint/v1/<API-KEY> -x '{"jsonrpc":"2.0", "id": 1, "method": "eth_unsubscribe", "params": ["0x9cef478923ff08bf67fde6c64013158d"]}'
```

5.4.2.25 eth_getStorageAt

简介

返回给定地址的存储位置的值。该API所消耗的计算单元为120。

参数说明

参数	类型	说明
地址	String	表示存储地址（20 字节）的字符串。
存储位置	String	存储位置的十六进制代码。
区块编号	String	想要查询的区块的十六进制的编号，或者是字符串"earliest"、"latest"、"pending"。

返回值

所提供存储位置的值。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  -d '{"jsonrpc":"2.0","method":"eth_getStorageAt","params":
["0x295a70b2de5e3953354a6a8344e616ed314d7251",
"0x6661e9d6d8b923d5bbaab1b96e1dd51ff6ea2a93520fdc9eb75d059238b8c5e9", "0x65a8db"],"id":1}'
```

5.4.2.26 eth_accounts

简介

返回客户端拥有的地址列表。该API所消耗的计算单元为103。

参数说明

此方法不接受任何参数。

返回值

数组，包含客户端拥有的地址的十六进制编码的字符串。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
    -X POST \
    -H "Content-Type: application/json" \
    -d '{"jsonrpc":"2.0","method":"eth_accounts","params":[],"id":1}'
```

5.4.2.27 eth_getCode

简介

返回给定地址处智能合约的已编译字节代码（如果有）。该API所消耗的计算单元为120。

参数说明

参数	类型	说明
地址	String	表示存储地址（20 字节）的字符串，从中获取编译后的字节码。
区块编码	String	想要查询的区块的十六进制的编号，或者是字符串"earliest"、"latest"、"pending"。

返回值

给定地址处智能合约的已编译字节代码。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  -d '{"jsonrpc":"2.0","method":"eth_getCode","params":
["0x06012c8cf97bead5deae237070f9587f8e7a266d", "0x65a8db"],"id":1}'
```

5.4.2.28 eth_getProof

简介

返回指定账户的账户和存储值，包括 Merkle 证明。该API所消耗的计算单元为120。

参数说明

参数	类型	说明
账户地址	String	表示存储地址（20 字节）的字符串，从中获取编译后的字节码。
存储键	Array	要验证和包含的 32 字节存储键值的数组。

参数	类型	说明
区块编号	String	想要查询的区块的十六进制的编号，或者是字符串"earliest"、"latest"、"pending"。

返回值

- address：与账户相关的地址。
- accountProof：RLP 序列化 MerkleTree-Nodes 的数组，从 stateRoot-Node 开始，遵循 SHA3（地址）的路径作为键。
- balance：当前余额的十六进制，以 wei 为单位。
- codeHash：账户代码的 32 字节哈希值。
- nonce：账户的随机数。
- storageHash: 32 字节。StorageRoot 的 SHA3。所有存储都将从这里开始提供 Merkle 证明rootHash。
- storageProof：请求的存储条目数组。每个条目都是一个具有以下属性的对象：
  - key：请求的存储密钥。
  - value：存储值。
  - proof：RLP 序列化 MerkleTree-Nodes 的数组，从 storageHash-Node 开始，遵循 SHA3（密钥）的路径作为路径。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  -d '{"jsonrpc": "2.0","method": "eth_getProof","id": 1,"params":
["0x7F0d15C7FAae65896648C8273B6d7E43f58Fa842",
["0x56e81f171bcc55a6ff8345e692c0f86e5b48e01b996cadc001622fb5e363b421"], "latest"]}'
```

5.4.2.29 eth_getLogs

简介

返回与给定过滤器对象匹配的所有日志的数组。该API所消耗的计算单元为120。

参数说明

参数	类型	说明
address	String	可选参数，合约地址（20字节）或日志应源自的地址列表。

参数	类型	说明
fromBlock	String	可选参数，默认为“latest”，十六进制区块号，或字符串latest,earliest或pending。
toBlock	String	可选参数，默认为“latest”，十六进制区块号，或字符串latest,earliest或pending。
topics	String	可选参数，32 字节 DATA 主题数组。主题与顺序相关。
blockhash	String	可选参数，将返回的日志限制为 32 字节哈希中引用的单个块blockHash。使用blockHash相当于设置fromBlock和 toBlock中引用的区块均为blockHash所对应的区块。如果blockHash出现在过滤条件中，则fromBlock和toBlock都不能够被设置。

返回值

日志对象数组，如果自上次轮询以来没有任何更改，则为空数组。日志对象包含以下键及其值：

- removed：若日志由于链重组而被删除，则返回true。如果它是有效的日志，则返回false。
- logIndex：块中日志索引位置的十六进制。当它是待处理（Pending）日志时返回NULL。
- transactionIndex：创建日志的事务索引位置的十六进制。当它是待处理（Pending）日志时返回NULL。
- transactionHash: 32 字节。创建此日志的事务的哈希值。当它是待处理（Pending）日志时返回NULL。
- blockHash: 32 字节。该日志所在块的哈希值，当它是待处理（Pending）日志时返回NULL。
- blockNumber：该日志所在的块号，当它是待处理（Pending）日志时返回NULL。
- address: 20 字节。该日志的来源地址。
- data：包含一个或多个 32 字节非索引日志参数。
- topics：包含 0 到 4 个索引日志参数的数组，每个 32 字节。在 Solidity 中，第一个主题是事件签名的哈希值（例如 Deposit(address,bytes32,uint256)），除非您使用匿名说明符声明事件。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  -d '{"jsonrpc":"2.0","method":"eth_getLogs","params":[{"blockHash":
"0x7c5a35e9cb3e8ae0e221ab470abae9d446c3a5626ce6689fc777dcffcab52c70", "topics":
["0x241ea03ca20251805084d27d4440371c34a0b85ff108f6bb5611248f73818b80"]}],"id":74}'
```

5.4.2.30 eth_getFilterChanges

简介

过滤器的轮询方法，返回自上次轮询以来发生的日志数组。过滤器必须通过调用eth_newFilter、eth_newBlockFilter、eth_newPendingTransactionFilter来创建。该API所消耗的计算单元为108。

参数说明

参数	类型	说明
过滤器ID	String	表示过滤器 ID 的字符串。

返回值

- log object array：（数组）日志对象数组，如果自上次轮询以来没有任何更改，则为空数组。
- 对于使用eth_newBlockFilter返回值创建的过滤器，返回值是块哈希（32 字节），例如["0x3454645634534..."]。
- 对于使用eth_newFilter日志创建的过滤器，对象具有以下参数：
  - address: 该日志的来源地址。
  - blockHash:该日志所在块的哈希值。当它是待处理（Pending）日志时返回NULL。
  - blockNumber：该日志所在的块号。当它是待处理（Pending）日志时返回NULL。
  - data： 包含日志的非索引参数。
  - logIndex：块中日志索引位置的十六进制。当它是待处理（Pending）日志时返回NULL。
  - removed：若日志由于链重组而被删除，则返回true。如果它是有效的日志，则返回false。
  - topics：数据数组。索引日志参数的 0 到 4 个 32 字节 DATA 的数组。在 Solidity 中，第一个topic是事件签名的哈希值（例如 Deposit(address,bytes32,uint256)），除非您使用匿名说明符声明事件。
  - transactionHash: 32 字节。创建此日志的事务的哈希值。当它是待处理（Pending）日志时返回NULL。
  - transactionIndex：创建此日志的事务索引位置的十六进制。当它是待处理（Pending）日志时返回NULL。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
   -X POST \
   -H "Content-Type: application/json" \
   -d '{"jsonrpc":"2.0","method":"eth_getFilterChanges","params":["0x16"],"id":73}'
```

5.4.2.31 eth_getFilterLogs

简介

返回与给定过滤器 ID 匹配的所有日志的数组。该API所消耗的计算单元为120。

参数说明

参数	类型	说明
address	String	可选参数，合约地址（20字节）或日志应源自的地址列表。
fromBlock	String	可选参数，默认为“latest”，十六进制区块号，或字符串latest、earliest或pending。
toBlock	String	可选参数，默认为“latest”，十六进制区块号，或字符串latest、earliest或pending。
topics	String	可选参数，32 字节 DATA 主题数组。主题与顺序相关。
blockhash	String	可选参数，将返回的日志限制为32字节哈希中引用的单个块blockHash。使用blockHash相当于设置fromBlock和 toBlock中引用的区块均为blockHash所对应的区块。如果blockHash出现在过滤条件中，则fromBlock和toBlock都不能够被设置。

返回值

- log 对象数组：与过滤器匹配的日志对象数组。对于自上次轮询以来发生的日志数组，请使用eth_getFilterChanges。日志对象包含以下键及其值：
  - address: 该日志的来源地址。
  - blockHash：该日志所在块的哈希值。当它是待处理（Pending）日志时返回NULL。
  - blockNumber：该日志所在的块号。当它是待处理（Pending）日志时返回NULL。
  - data：包含日志的非索引参数。
  - logIndex：块中日志索引位置的十六进制。当它是待处理（Pending）日志时返回NULL。
  - removed：若日志由于链重组而被删除，则返回true。如果它是有效的日志，则返回false。
  - topics：数据数组。索引日志参数的 0 到 4 个 32 字节 DATA 的数组。在 Solidity 中，第一个topic是事件签名的哈希值（例如 Deposit(address、bytes32、uint256)），除非您使用匿名说明符声明事件。

- transactionHash: 创建此日志的事务的哈希值。当它是待处理（Pending）日志时返回NULL。
- transactionIndex：创建此日志的事务索引位置的十六进制。当它是待处理（Pending）日志时返回NULL。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
   -X POST \
   -H "Content-Type: application/json" \
   -d '{"jsonrpc":"2.0","method":"eth_getFilterLogs","params":["0x16"],"id":74}'
```

5.4.2.32 eth_newBlockFilter

简介

在节点中创建一个过滤器，以在新区块到达时发出通知。该API所消耗的计算单元为144。

参数说明

此方法不接受任何参数。

返回值

十六进制表示的新创建的过滤器ID。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
   -X POST \
   -H "Content-Type: application/json" \
   -d '{"jsonrpc":"2.0","method":"eth_newBlockFilter","params":[],"id":73}'
```

5.4.2.33 eth_newFilter

简介

根据给定的过滤器选项创建过滤器对象，以在状态更改（日志）时发出通知。该API所消耗的计算单元为120。

参数说明

参数	类型	说明
address	String	可选参数，合约地址（20字节）或日志应源自的地址列表。
fromBlock	String	可选参数，默认为“latest”，十六进制区块号，或字符串latest，earliest或pending。

参数	类型	说明
toBlock	String	可选参数，默认为“latest”，十六进制区块号，或字符串latest，earliest或pending。
topics	String	可选参数，32 字节 DATA 主题数组。主题与顺序相关。

返回值

十六进制表示的新创建的过滤器ID。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
   -X POST \
   -H "Content-Type: application/json" \
   -d '{"jsonrpc":"2.0","method":"eth_newFilter","params":[{"topics":
["0xddf252ad1be2c89b69c2b068fc378daa952ba7f163c4a11628f55a4df523b3ef"]}],"id":73}'
```

5.4.2.34 eth_newPendingTransactionFilter

简介

在节点中创建一个过滤器，以在新的待处理事务到达BNB Smart Chain时发出通知。该API所消耗的计算单元为252。

参数说明

此方法不接受任何参数。

返回值

十六进制表示的新创建的过滤器ID。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
   -X POST \
   -H "Content-Type: application/json" \
   -d '{"jsonrpc":"2.0","method":"eth_newPendingTransactionFilter","params":[],"id":73}'
```

5.4.2.35 eth_uninstallFilter

简介

卸载具有给定 id 的过滤器。当不再需要观察时调用。此外，如果一段时间内没有通过eth_getFilterChanges请求过滤器，过滤器就会超时。该API所消耗的计算单元为127。

参数说明

参数	类型	说明
过滤器ID	String	要卸载的过滤器ID的字符串。

返回值

如果过滤器已成功卸载，返回true，否则false。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
    -X POST \
    -H "Content-Type: application/json" \
    -d '{"jsonrpc":"2.0","method":"eth_uninstallFilter","params":["0xb"],"id":73}'
```

5.4.2.36 eth_chainId

简介

返回当前配置的链 ID。该API所消耗的计算单元为103。

参数说明

此方法不接受任何参数。

返回值

十六进制表示的链 ID。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
    -X POST \
    -H "Content-Type: application/json" \
    -d '{"jsonrpc":"2.0","method":"eth_chainId","params": [],"id":1}'
```

5.4.2.37 web3_sha3

简介

返回给定数据的 Keccak-256 编码结果（不是标准化 SHA3-256 ）。该API所消耗的计算单元为131。

参数说明

参数	类型	说明
数据	String	需要转换的数据

返回值

给定输入的SHA3编码后的结果。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
   -X POST \
   -H "Content-Type: application/json" \
   -d '{"jsonrpc":"2.0","method":"web3_sha3","params": ["0x68656c6c6f20776f726c64"],"id":64}'
```

5.4.2.38 web3_clientVersion

简介

返回当前客户端的版本。该API所消耗的计算单元为137。

参数说明

此方法不接受任何参数。

返回值

客户端的版本。

5.4.2.39 txpool_status

简介

返回处于待处理（Pending）状态和排队（Queued）状态的事务数。该API所消耗的计算单元为120。

参数说明

此方法不接受任何参数。

返回值

Object - 交易对象包含以下字段：

- pending: 交易池里处于待处理（Pending）状态的交易总数，16进制表示。
- queued: 交易池里处于待处理（queued）状态的交易总数，16进制表示。

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"txpool_status","params":[],"id":1,"jsonrpc":"2.0"}'
```

5.4.2.40 net_listening

简介

当客户端正在主动侦听网络连接时为true。该API所消耗的计算单元为130。

参数说明

此方法不接受任何参数。

返回值

当客户端正在主动侦听网络连接时为true，否则为false

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"net_listening","params":[],"id":1,"jsonrpc":"2.0"}'
```

5.4.2.41 net_version

简介

返回当前网络id。该API所消耗的计算单元为107。

参数说明

此方法不接受任何参数。

返回值

Object - 当前网络id的字符串值。典型值如下：

- 1 - ethereum mainnet
- 2 - morden testnet (deprecated)
- 3 - ropsten testnet
- 4 - rinkeby testnet
- 5 - goerli testnet
- 11155111 - sepolia testnet
- 10 - optimism mainnet
- 69 - optimism kovan testnet
- 42 - kovan testnet
- 137 - matic/polygon mainnet
- 80001 - matic/polygon mumbai testnet
- 250 - fantom mainnet
- 100 - xdai mainnet
- 56 - bsc mainnet

请求样式

```
curl https://your-http-endpoint/v1/<API-KEY> \
  -X POST \
  -H "Content-Type: application/json" \
  --data '{"method":"net_version","params":[],"id":1,"jsonrpc":"2.0"}'
```

6 批量请求

6.1 批量请求介绍

批量请求是通过单个HTTP请求包含多个嵌套的API调用。客户端可以同时发送多个请求，填充在一个数组中，服务端会返回相应的响应数组。

服务端在用户套餐包对应的计算单元限制内会并发地处理此批RPC调用的所有请求。超出计算单元限制的RPC调用会等待前面的调用处理完成后继续处理。

包含不同RPC调用的批量请求很容易变得复杂，与单个API调用相比，批处理请求的可靠性更低，因此不建议使用批量请求。

6.2 批量请求范围

支持节点引擎服务各公链已开放的所有HTTP JSON-RPC接口批量调用，WebSocket接口暂不支持。

6.3 批量请求示例

以eth_getFilterChange接口为例，批量请求和单个请求的区别是请求body以数组形式包装多个子请求。服务端以数组形式包装返回各子请求的响应。

单个请求示例

```
curl https://your-http-endpoint/v1/<API-KEY> \
   -X POST \
   -H "Content-Type: application/json" \
   -d '{"jsonrpc":"2.0","method":"eth_getFilterChanges","params":["0x16"],"id":1}'
```

批量请求示例

```
curl https://your-http-endpoint/v1/<API-KEY> \
   -X POST \
   -H "Content-Type: application/json" \
   -d '[{"jsonrpc":"2.0","method":"eth_getFilterChanges","params":["0x16"],"id":1},
{"jsonrpc":"2.0","method":"eth_getFilterChanges","params":["0x16"],"id":2},
{"jsonrpc":"2.0","method":"eth_getFilterChanges","params":["0x16"],"id":3},
{"jsonrpc":"2.0","method":"eth_getFilterChanges","params":["0x16"],"id":4}]'
```

返回值

各个RPC调用对应的响应数组。

```
[{"jsonrpc":"2.0","id":1,"result":null},{"jsonrpc":"2.0","id":2,"result":null},{"jsonrpc":"2.0","id":3,"result":null},
{"jsonrpc":"2.0","id":4,"result":null}]
```

7 增强 API

7.1 增强 API 介绍

节点引擎服务为用户提供增强API。当前已支持Gas优化相关API。

7.2 增强 API 列表

7.2.1 Gas 优化 API

Gas优化API可以在不影响交易成功率和确认速率的情况下，节约Gas成本。

说明

当前仅专享版支持Gas优化API。

Gas 优化流程

开发者可以向节点发送一组Gas不同的裸交易(rawTransaction)，Gas优化API将会在后台尝试以下操作：

步骤1 将这批交易根据Gas从低到高顺序排列。

步骤2 提交Gas最低的交易。

步骤3 检查交易是否已经被区块确认。

步骤4 如果交易在4s内未被确认，则会尝试提交下一个Gas更高的交易，直至交易被确认。

----结束

注意事项

1. 当前Gas优化API支持两种模式：EIP1559和传统交易。
   - 采用EIP1559交易时，一组交易除了maxPriorityFeePerGas不同，其他参数应保证完全一致；
   - 采用传统交易时，一组交易除了gasPrice不同，其他参数应保持完全一致。

2. 同一组交易不能同时包括EIP1559和传统交易两种模式。
3. Gas优化API仅支持适配eth_sendRawTransactionAPI接口的链：Ethereum、Arbitrum、Polygon、BSC。

7.2.1.1 nes_sendGasOptimizedTransaction

nes_sendGasOptimizedTransation接收入参为string类型的裸交易数组，并返回用于查询该Gas优化交易状态的id。接口接收裸交易数组长度限制为[1,10]。

请求示例

```
curl https://your-http-endpoint/your-credential \
   -X POST \
   -H "Content-Type: application/json" \
   -d '{"jsonrpc":"2.0","method":"nes_sendGasOptimizedTransation","params":[
"0x02f87583064aba048405f5e10085012a05f20082520894958a15271aa13f6b7feb029b6114a69e6de8b693872386f26fc1000080c001a0df9b7a8ab18f081930b6c6e85fd75fdbaa1de8e0027f21bff1aeeb9d6ff6e477a062bec3ad68abc739c3bc96305a97d7952627cd95190f86e4f0cb3ffbcc002623",
"0x02f87483064aba048405f5e10084ee6b280082520894958a15271aa13f6b7feb029b6114a69e6de8b693872386f26fc1000080c001a024b016079f1aa5b437f5f8c7aee25a3accb0873eab7ac86f39af10068ee725efa05896d1f7562f21f185b23d4d6c0cb735518d3e5c9f35d5fa5f900b6b9b0f80d9",
"0x02f87483064aba048405f5e10084b2d05e0082520894958a15271aa13f6b7feb029b6114a69e6de8b693872386f26fc1000080c080a00f3767650d09f1330953abd4498fd8bca3ca0c1444cf56ca67e4ebda003d5ba4a042d961648b42b33846c4e1812b5727c1fba4f9cad8602b708251c912aa3f647e"
],"id":1}'
```

返回值

Gas优化交易的Id，可作为nes_getGasOptimizedTransactionStatus的入参传入，获取该组增强交易的状态。

```
{"jsonrpc":"2.0","id":1,"result":"0x4201"}
```

7.2.1.2 nes_getGasOptimizedTransactionStatus

nes_getGasOptimizedTransactionStatus接收入参为Gas优化交易的id，返回增强交易的状态。

请求示例

```
curl https://your-http-endpoint/your-credential \
   -X POST \
   -H "Content-Type: application/json" \
   -d '{"jsonrpc":"2.0","method":"nes_getGasOptimizedTransactionStatus","params":["0x4201"],"id":1}'
```

返回值

返回值可能有pending、success、failed三种状态：

1. 当接口被调用时执行时，Gas优化会进入pending状态。
2. 如果交易一直未被确认，Gas优化交易也会保持pending状态。
3. 当交易被确认后会进入success状态。
4. 如果执行流程中出现错误，则会进入failed状态。

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