

# JSON-RPC methods

Here you can find the JSON-RPC API endpoints. You [can call these APIs using a variety of tools](#).

## Error codes

The following list contains all possible error codes and associated messages:

Code Message Meaning Category -32700 Parse error Invalid JSON standard -32600 Invalid request JSON is not a valid request object standard -32601 Method not found Method does not exist standard -32602 Invalid params Invalid method parameters standard -32603 Internal error Internal JSON-RPC error standard -32000 Invalid input Missing or invalid parameters non-standard -32001 Resource not found Requested resource not found non-standard -32002 Resource unavailable Requested resource not available non-standard -32003 Transaction rejected Transaction creation failed non-standard -32004 Method not supported Method is not implemented non-standard -32005 Limit exceeded Request exceeds defined limit non-standard -32006 JSON-RPC version not supported Version of JSON-RPC protocol is not supported non-standard Example error response:

```
{ "id": 1337, "jsonrpc": "2.0", "error": { "code": -32003, "message": "Transaction rejected" } }
```

## Value encoding

Specific types of values passed to and returned from Ethereum RPC methods require special encoding:

### Quantity

AQuantity (integer, number) must:

- Be hex-encoded.
- Be "0x"-prefixed.
- Be expressed using the fewest possible hex digits per byte.
- Express zero as "0x0".

ExamplesQuantity values:

Value Validity Reason 0x invalid empty not a valid quantity 0x0 valid interpreted as a quantity of zero 0x00 invalid leading zeroes not allowed 0x41 valid interpreted as a quantity of 65 0x400 valid interpreted as a quantity of 1024 0x0400 invalid leading zeroes not allowed ff invalid values must be prefixed

### Block identifier

The RPC methods below take a default block identifier as a parameter.

- eth\_getBalance
- eth\_getStorageAt
- eth\_getTransactionCount
- eth\_getCode
- eth\_call
- eth\_getProof

Since there is no way to clearly distinguish between aData parameter and aQuantity parameter [EIP-1898](#) provides a format to specify a block either using the block hash or block number. The block identifier is a JSONObject with the following fields:

Property Type Description [blockNumber] {Quantity} The block in the canonical chain with this number OR[blockHash] {Data} The block uniquely identified by this hash. TheblockNumber andblockHash properties are mutually exclusive; exactly one of them must be set. requireCanonical {boolean} (optional) Whether or not to throw an error if the block is not in the canonical chain as described below. Only allowed in conjunction with theblockHash tag. Defaults tofalse .

### Data

AData value (for example, byte arrays, account addresses, hashes, and bytecode arrays) must:

- Be hex-encoded.
- Be "0x"-prefixed.
- Be expressed using two hex digits per byte.

ExamplesData values:

Value Valid Reason 0x valid interpreted as empty data 0x0 invalid each byte must be represented using two hex digits 0x00

valid interpreted as a single zero byte 0x41 true interpreted as a data value of 65 0x004200 true interpreted as a data value of 16896 0xf0f0f false bytes require two hex digits 004200 false values must be prefixed

Last updated on Apr 19, 2024 [Previous](#) [Trace transactions](#) [Next](#) [eth\\_accounts](#)