

# Package ‘ether’

September 22, 2018

**Type** Package

**Title** Interaction with the 'Ethereum' Blockchain

**Version** 0.1.5

**Description**

Interacts with the open-source, public 'Ethereum' <<https://www.ethereum.org/>> blockchain. It provides a distributed computing platform via smart contracts. This package provides functions which interrogate blocks and transactions in the 'Ethereum' blockchain.

**License** GPL-2

**Encoding** UTF-8

**LazyData** true

**Imports** dplyr, httr, jsonlite, Rmpfr

**RoxygenNote** 6.1.0

**Suggests** testthat

**NeedsCompilation** no

**Author** Andrew Collier [aut, cre]

**Maintainer** Andrew Collier <[andrew@exegetic.biz](mailto:andrew@exegetic.biz)>

**Repository** CRAN

**Date/Publication** 2018-09-22 14:10:03 UTC

## R topics documented:

as.ether	2
as.finney	3
as.gwei	3
as.kwei	3
as.mwei	4
as.szabo	4
dec_to_hex	4
ether	5
eth_accounts	5
eth_blockNumber	6

eth_coinbase	6
eth_gasPrice	7
eth_getBalance	7
eth_getBlock	8
eth_getBlockTransactionCountByHash	9
eth_getBlockTransactionCountByNumber	9
eth_getStorageAt	10
eth_getTransactionByBlockHashAndIndex	11
eth_getTransactionByBlockNumberAndIndex	11
eth_getTransactionByHash	12
eth_getTransactionCount	13
eth_getTransactionReceipt	13
eth_getUncleByBlockHashAndIndex	14
eth_getUncleByBlockNumberAndIndex	15
eth_getUncleCountByBlockHash	15
eth_getUncleCountByBlockNumber	16
eth_hashrate	17
eth_mining	17
eth_protocolVersion	18
eth_syncing	18
get_blocks	19
get_rpc_address	19
get_transactions	20
hex_to_dec	20
set_rpc_address	21
web3_clientVersion	21
web3_sha3	22

## Index 23

---

as.ether	<i>Convert Wei to Ether.</i>
----------	------------------------------

---

### Description

Convert Wei to Ether.

### Usage

```
as.ether(wei)
```

### Arguments

wei	Number of Wei.
-----	----------------

---

as.finney	<i>Convert Wei to Finney.</i>
-----------	-------------------------------

---

**Description**

Convert Wei to Finney.

**Usage**

as.finney(wei)

**Arguments**

wei	Number of Wei.
-----	----------------

---

as.gwei	<i>Convert Wei to GWei.</i>
---------	-----------------------------

---

**Description**

Convert Wei to GWei.

**Usage**

as.gwei(wei)

**Arguments**

wei	Number of Wei.
-----	----------------

---

as.kwei	<i>Convert Wei to kWei.</i>
---------	-----------------------------

---

**Description**

Convert Wei to kWei.

**Usage**

as.kwei(wei)

**Arguments**

wei	Number of Wei.
-----	----------------

---

as.mwei	<i>Convert Wei to MWei.</i>
---------	-----------------------------

---

**Description**

Convert Wei to MWei.

**Usage**

```
as.mwei(wei)
```

**Arguments**

wei	Number of Wei.
-----	----------------

---

as.szabo	<i>Convert Wei to Szabo.</i>
----------	------------------------------

---

**Description**

Convert Wei to Szabo.

**Usage**

```
as.szabo(wei)
```

**Arguments**

wei	Number of Wei.
-----	----------------

---

dec_to_hex	<i>Convert decimal to hexadecimal.</i>
------------	--

---

**Description**

Convert decimal to hexadecimal.

**Usage**

```
dec_to_hex(dec)
```

**Arguments**

dec	Decimal number.
-----	-----------------

**Value**

String.

**Examples**

```
dec_to_hex(21106)
```

---

ether	<i>ether: An R package for working with Ethereum.</i>
-------	---

---

**Description**

The ether package provides an interface to the Ethereum RPC functions.

---

eth_accounts	<i>Returns addresses owned by client.</i>
--------------	---

---

**Description**

Returns addresses owned by client.

**Usage**

```
eth_accounts()
```

**Value**

A vector of 20 byte (hexadecimal) Ethereum addresses.

**Examples**

```
## Not run:  
eth_accounts()  
  
## End(Not run)
```

---

eth_blockNumber	<i>Returns the number of most recent block.</i>
-----------------	---

---

**Description**

Returns the number of most recent block.

**Usage**

```
eth_blockNumber()
```

**Value**

Integer.

**Examples**

```
## Not run:  
eth_blockNumber()  
  
## End(Not run)
```

---

eth_coinbase	<i>Returns the client coinbase address.</i>
--------------	---

---

**Description**

Returns the client coinbase address.

**Usage**

```
eth_coinbase()
```

**Value**

A 20 byte (hexadecimal) Ethereum address.

**Examples**

```
## Not run:  
eth_coinbase()  
  
## End(Not run)
```

---

eth_gasPrice	<i>Returns the current gas price in wei.</i>
--------------	--

---

**Description**

Returns the current gas price in wei.

**Usage**

```
eth_gasPrice()
```

**Value**

Integer.

**Examples**

```
## Not run:  
eth_gasPrice()  
  
## End(Not run)
```

---

eth_getBalance	<i>Returns the balance (in Wei) of the account at specified address.</i>
----------------	--

---

**Description**

Returns the balance (in Wei) of the account at specified address.

**Usage**

```
eth_getBalance(address = NULL, number = "latest")
```

**Arguments**

address	A 20 byte (hexadecimal) Ethereum address.
number	An integer block number (as a hexadecimal string) or one of "earliest", "latest" or "pending".

**Value**

Integer.

**Examples**

```
## Not run:
eth_getBalance()
eth_getBalance("0xD34DA389374CAAD1A048FBDC4569AAE33FD5a375")

## End(Not run)
```

---

eth_getBlock	<i>Returns information about a specified block.</i>
--------------	---

---

**Description**

Combines functionality from eth\_getBlockByHash() and eth\_getBlockByNumber().

**Usage**

```
eth_getBlock(hash = NULL, number = "latest", full = TRUE)
```

**Arguments**

hash	Hash of a block.
number	An integer block number (as a hexadecimal string) or one of "earliest", "latest" or "pending".
full	Whether or not to return full transaction objects.

**Value**

List.

**Examples**

```
## Not run:
eth_getBlock()
eth_getBlock("0xb6d656ead4c3d4b1aa24d6b4d3d4cde8c090794e597258993512d650f088fcba")
eth_getBlock(number = "0x4720FF")
eth_getBlock(number = "0x49a8ea") # A block with two uncles.

## End(Not run)
```



---

eth\_getBlockTransactionCountByHash

*Returns the number of transactions in a block matching the given block hash.*

---

### Description

Returns the number of transactions in a block matching the given block hash.

### Usage

```
eth_getBlockTransactionCountByHash(hash)
```

### Arguments

hash	Hash of a block.
------	------------------

### Value

Integer.

### Examples

```
## Not run:
eth_getBlockTransactionCountByHash(
  "0xb6d656ead4c3d4b1aa24d6b4d3d4cde8c090794e597258993512d650f088fcb"
)
## End(Not run)
```

---

eth\_getBlockTransactionCountByNumber

*Returns the number of transactions in a specified block.*

---

### Description

Returns the number of transactions in a specified block.

### Usage

```
eth_getBlockTransactionCountByNumber(number = "latest")
```

### Arguments

number	An integer block number (as a hexadecimal string) or one of "earliest", "latest" or "pending".
--------	--

**Value**

Integer.

**Examples**

```
## Not run:  
eth_getBlockTransactionCountByNumber("0x4720FF")  
eth_getBlockTransactionCountByNumber()  
  
## End(Not run)
```

---

eth_getStorageAt	<i>Returns the value from a storage position at a given address.</i>
------------------	--

---

**Description**

Returns the value from a storage position at a given address.

**Usage**

```
eth_getStorageAt(address, position, number = "latest")
```

**Arguments**

address	A 20 byte (hexadecimal) Ethereum address.
position	Position in the storage.
number	An integer block number (as a hexadecimal string) or one of "earliest", "latest" or "pending".

**Value**

Hex string.

**Examples**

```
## Not run:  
eth_getStorageAt("0x3589d05a1ec4Af9f65b0E5554e645707775Ee43C", "0x0")  
  
## End(Not run)
```

---

eth\_getTransactionByBlockHashAndIndex

*Returns information about a transaction by block hash and transaction index position.*

---

### Description

Returns information about a transaction by block hash and transaction index position.

### Usage

```
eth_getTransactionByBlockHashAndIndex(hash, index)
```

### Arguments

hash	A block hash.
index	Index of transaction in block.

### Value

List.

### Examples

```
## Not run:
eth_getTransactionByBlockHashAndIndex(
    "0xb6d656ead4c3d4b1aa24d6b4d3d4cde8c090794e597258993512d650f088fcb", "0x0"
)
eth_getTransactionByBlockHashAndIndex(
    "0xb6d656ead4c3d4b1aa24d6b4d3d4cde8c090794e597258993512d650f088fcb", "0x5"
)

## End(Not run)
```

---

eth\_getTransactionByBlockNumberAndIndex

*Returns information about a transaction by block number and transaction index position.*

---

### Description

Returns information about a transaction by block number and transaction index position.

### Usage

```
eth_getTransactionByBlockNumberAndIndex(number, index)
```

**Arguments**

number	An integer block number (as a hexadecimal string) or one of "earliest", "latest" or "pending".
index	Index of transaction in block.

**Value**

List.

**Examples**

```
## Not run:
eth_getTransactionByBlockNumberAndIndex("0x4720FF", "0x0")
eth_getTransactionByBlockNumberAndIndex("0x4720FF", "0x5")

## End(Not run)
```

---

eth\_getTransactionByHash

*Returns the information about a transaction.*

---

**Description**

Returns the information about a transaction.

**Usage**

```
eth_getTransactionByHash(hash)
```

**Arguments**

hash	A transaction hash.
------	---------------------

**Value**

List.

**Examples**

```
## Not run:
eth_getTransactionByHash("0x4b0f317c326299a4d063ccccff2daf7452c35dc7ec5b4279f001d6e2f745cc6f")
eth_getTransactionByHash("0x8fd9e04958cc4fb602b9d9fa5a9b6da512779ccd22477ccc5ce73721296cf151")

## End(Not run)
```

---

`eth_getTransactionCount`*Returns the number of transactions sent from an address.*

---

**Description**

Returns the number of transactions sent from an address.

**Usage**

```
eth_getTransactionCount(address = NULL, number = "latest")
```

**Arguments**

address	A 20 byte (hexadecimal) Ethereum address.
number	An integer block number (as a hexadecimal string) or one of "earliest", "latest" or "pending".

**Value**

Integer.

**Examples**

```
## Not run:  
eth_getTransactionCount("0xD34DA389374CAAD1A048FBDC4569AAE33FD5a375")  
  
## End(Not run)
```

---

`eth_getTransactionReceipt`*Returns the receipt of a transaction by transaction hash.*

---

**Description**

Returns the receipt of a transaction by transaction hash.

**Usage**

```
eth_getTransactionReceipt(hash)
```

**Arguments**

hash	A transaction hash.
------	---------------------

**Value**

List.

**Examples**

```
## Not run:
eth_getTransactionReceipt("0x8fd9e04958cc4fb602b9d9fa5a9b6da512779ccd22477ccc5ce73721296cf151")

## End(Not run)
```

---

eth\_getUncleByBlockHashAndIndex

*Returns information about an uncle of a block by hash and uncle index position.*

---

**Description**

Returns information about an uncle of a block by hash and uncle index position.

**Usage**

```
eth_getUncleByBlockHashAndIndex(hash, index)
```

**Arguments**

hash	Hash of a block.
index	Index of transaction in block.

**Value**

List.

**Examples**

```
## Not run:
eth_getUncleByBlockHashAndIndex(
    "0x8575df1eb3df61f3880628ca8e495038ee0b278c0aa48fe41f80b8d0d4e83e79", "0x0"
)

## End(Not run)
```

---

`eth_getUncleByBlockNumberAndIndex`

*Returns information about a uncle of a block by number and uncle index position.*

---

**Description**

Returns information about a uncle of a block by number and uncle index position.

**Usage**

```
eth_getUncleByBlockNumberAndIndex(number, index)
```

**Arguments**

number	An integer block number (as a hexadecimal string) or one of "earliest", "latest" or "pending".
index	Index of transaction in block.

**Value**

List.

**Examples**

```
## Not run:  
eth_getUncleByBlockNumberAndIndex("0x4720fe", "0x0")  
  
## End(Not run)
```

---

`eth_getUncleCountByBlockHash`

*Returns the number of uncles in a block from a block matching the given block hash.*

---

**Description**

Returns the number of uncles in a block from a block matching the given block hash.

**Usage**

```
eth_getUncleCountByBlockHash(hash)
```

**Arguments**

hash	Hash of a block.
------	------------------

**Value**

Integer.

**Examples**

```
## Not run:  
eth_getUncleCountByBlockHash("0x8575df1eb3df61f3880628ca8e495038ee0b278c0aa48fe41f80b8d0d4e83e79")  
  
## End(Not run)
```

---

eth\_getUncleCountByBlockNumber

*Returns the number of uncles in a block from a block matching the given block number.*

---

**Description**

Returns the number of uncles in a block from a block matching the given block number.

**Usage**

```
eth_getUncleCountByBlockNumber(number)
```

**Arguments**

number	An integer block number (as a hexadecimal string) or one of "earliest", "latest" or "pending".
--------	--

**Value**

Integer.

**Examples**

```
## Not run:  
eth_getUncleCountByBlockNumber("0x4720fe")  
  
## End(Not run)
```



---

eth_hashrate	<i>Returns the number of hashes per second that the node is mining with.</i>
--------------	--

---

**Description**

Returns the number of hashes per second that the node is mining with.

**Usage**

```
eth_hashrate()
```

**Value**

Integer.

**Examples**

```
## Not run:  
eth_hashrate()  
  
## End(Not run)
```

---

eth_mining	<i>Returns true if client is actively mining new blocks.</i>
------------	--

---

**Description**

Returns true if client is actively mining new blocks.

**Usage**

```
eth_mining()
```

**Value**

Boolean.

**Examples**

```
## Not run:  
eth_mining()  
  
## End(Not run)
```

eth\_protocolVersion     *Returns the current ethereum protocol version.*

---

**Description**

Returns the current ethereum protocol version.

**Usage**

```
eth_protocolVersion()
```

**Value**

Integer.

**Examples**

```
## Not run:  
eth_protocolVersion()  
  
## End(Not run)
```

---

eth\_syncing             *Returns an object with data about the sync status.*

---

**Description**

Returns an object with data about the sync status.

**Usage**

```
eth_syncing()
```

**Value**

Boolean.

**Examples**

```
## Not run:  
eth_syncing()  
  
## End(Not run)
```

---

get_blocks	<i>Retrieve a series of blocks.</i>
------------	-------------------------------------

---

**Description**

Retrieve a series of blocks.

**Usage**

```
get_blocks(start = NULL, stop = NULL, count = NULL)
```

**Arguments**

start	Initial block number.
stop	Final block number.
count	Number of blocks.

**Value**

Data frame of blocks.

**Examples**

```
## Not run:  
get_blocks("0x4720FF", "0x472108")  
get_blocks("0x4720FF", count = 5)  
get_blocks("0x49a8ea", count = 5) # Two blocks containing uncles.  
  
## End(Not run)
```

---

get_rpc_address	<i>Get address used for RPC.</i>
-----------------	----------------------------------

---

**Description**

Get address used for RPC.

**Usage**

```
get_rpc_address()
```

**Value**

URL.

---

get_transactions	<i>Retrieve transactions for a series of blocks.</i>
------------------	--

---

**Description**

Retrieve transactions for a series of blocks.

**Usage**

```
get_transactions(start = NULL, stop = NULL, count = NULL)
```

**Arguments**

start	Initial block number.
stop	Final block number.
count	Number of blocks.

**Value**

Data frame of transactions.

**Examples**

```
## Not run:  
get_transactions("0x4720FF", "0x472108")  
get_transactions("0x4720FF", count = 5)  
  
## End(Not run)
```

---

hex_to_dec	<i>Convert hexadecimal to decimal.</i>
------------	--

---

**Description**

Convert hexadecimal to decimal.

**Usage**

```
hex_to_dec(hex, integer = TRUE)
```

**Arguments**

hex	Hexadecimal number as string.
integer	Whether result should be an integer.

**Value**

Numeric or integer.

**Examples**

```
hex_to_dec("0x5272")
```

---

set_rpc_address	<i>Set address used for RPC.</i>
-----------------	----------------------------------

---

**Description**

Set address used for RPC.

**Usage**

```
set_rpc_address(url, port = NULL, key = NULL)
```

**Arguments**

url	URL for RPC endpoint.
port	Port number for RPC connection.
key	Connection key.

**Examples**

```
set_rpc_address("http://localhost:8545")
```

---

web3_clientVersion	<i>Returns the current client version.</i>
--------------------	--

---

**Description**

Returns the current client version.

**Usage**

```
web3_clientVersion()
```

**Value**

String.

**Examples**

```
## Not run:
web3_clientVersion()

## End(Not run)
```

---

web3_sha3	<i>Returns Keccak-256 (not the standardized SHA3-256) of the given data.</i>
-----------	--

---

**Description**

Returns Keccak-256 (not the standardized SHA3-256) of the given data.

**Usage**

```
web3_sha3(data)
```

**Arguments**

data	Data encoded as hex.
------	----------------------

**Value**

String.

**Examples**

```
## Not run:  
web3_sha3("0x68656c6c6f20776f726c64")  
  
## End(Not run)
```

# Index

as.ether, [2](#)  
as.finney, [3](#)  
as.gwei, [3](#)  
as.kwei, [3](#)  
as.mwei, [4](#)  
as.szabo, [4](#)

dec\_to\_hex, [4](#)

eth\_accounts, [5](#)  
eth\_blockNumber, [6](#)  
eth\_coinbase, [6](#)  
eth\_gasPrice, [7](#)  
eth\_getBalance, [7](#)  
eth\_getBlock, [8](#)  
eth\_getBlockTransactionCountByHash, [9](#)  
eth\_getBlockTransactionCountByNumber,  
[9](#)  
eth\_getStorageAt, [10](#)  
eth\_getTransactionByBlockHashAndIndex,  
[11](#)  
eth\_getTransactionByBlockNumberAndIndex,  
[11](#)  
eth\_getTransactionByHash, [12](#)  
eth\_getTransactionCount, [13](#)  
eth\_getTransactionReceipt, [13](#)  
eth\_getUncleByBlockHashAndIndex, [14](#)  
eth\_getUncleByBlockNumberAndIndex, [15](#)  
eth\_getUncleCountByBlockHash, [15](#)  
eth\_getUncleCountByBlockNumber, [16](#)  
eth\_hashrate, [17](#)  
eth\_mining, [17](#)  
eth\_protocolVersion, [18](#)  
eth\_syncing, [18](#)  
ether, [5](#)  
ether-package (ether), [5](#)

get\_blocks, [19](#)  
get\_rpc\_address, [19](#)  
get\_transactions, [20](#)

hex\_to\_dec, [20](#)

set\_rpc\_address, [21](#)

web3\_clientVersion, [21](#)  
web3\_sha3, [22](#)