**aiman.js 使用和接口文档**

# 安装使用

## npm install aiman

使用示例:

*//初始化过程*

**var** AIMan = require('AIMan');

**if** (**typeof** aiman !== 'undefined') {

aiman = **new** AIMan(aiman.currentProvider);

} **else** {

*// set the provider you want from AIMan.providers*

aiman = **new** AIMan(**new** AIMan.providers.HttpProvider("http://localhost:8545"));

}

# 部分接口说明

## aiman.version.network

同步方式：aiman.version.network

异步方式：aiman.version.getNetwork(callback(error, result){ ... })

返回值:

String - 网络协议版本

示例：

*//省略初始化过程*

**var** version = aiman.version.network;

console.log(version);

## aiman.setProvider

aiman.setProvider

设置Provider

参数：无

返回值：undefined

示例：

aiman.setProvider(**new** aiman.providers.HttpProvider('http://localhost:8545'));

## aiman.sha3

aiman.sha3(string, options)

参数：

String：String - 传入的需要使用Keccak-256 SHA3算法进行哈希运算的字符串。

options：Object - 可选项设置。如果要解析的是hex格式的十六进制字符串。需要设置encoding为hex。因为JS中会默认忽略0x。

返回值：

String - 使用Keccak-256 SHA3算法哈希过的结果。

示例：

//省略初始化过程

var hash = aiman.sha3("Some string to be hashed");

console.log(hash);

var hashOfHash = aiman.sha3(hash, {encoding: 'hex'});

console.log(hashOfHash);

## aiman.toHex

aiman.toHex

将任何值转为HEX。

参数：

String|Number|Object|Array|BigNumber - 需要转化为HEX的值。如果是一个对象或数组类型，将会先用JSON.stringify进行转换成字符串。如果传入的是BigNumber，则将得到对应的Number的HEX。

返回值：

16进制字符串

示例：

//初始化基本对象

var AIMan = require('aiman');

var aiman = **new** AIMan(**new** AIMan.providers.HttpProvider("http://localhost:8545"));

var BigNumber = require('bignumber.js'); //version 5.0.0

var str = "abcABC";

var obj = {abc: 'ABC'};

var bignumber = **new** BigNumber('12345678901234567890');

var hstr = aiman.toHex(str);

var hobj = aiman.toHex(obj);

var hbg = aiman.toHex(bignumber);

console.log("Hex of Sring:" + hstr);

console.log("Hex of Object:" + hobj);

console.log("Hex of BigNumber:" + hbg);

$ node test.js

Hex **of** Sring:0x616263414243

Hex **of** Object:0x7b22616263223a22414243227d

Hex **of** BigNumber:0xab54a98ceb1f0ad2

## aiman.toAscii

aiman.toAscii(hexString)

将HEX字符串转为ASCII[3](http://web3.tryblockchain.org/Web3.js-api-refrence.html#fn3)字符串

参数：

1. String - 十六进制字符串。

返回值：

String - 给定十六进制字符串对应的ASCII码值。

## aiman.fromAscii

aiman.fromAscii

将任何的ASCII码字符串转为HEX字符串。

参数：

String - ASCII码字符串

Number - 返回的字符串字节大小，不够长会自动填充。

返回值：String - 转换后的HEX字符串。

示例：

**var** str = aiman.fromAscii('matrix');

console.log(str); *//*

**var** str2 = aiman.fromAscii('matrix', 32);

console.log(str2); *//*

## aiman.toDecimal

aiman.toDecimal

将一个十六进制转为一个十进制的数字

参数：String - 十六进制字符串

返回：Number - 传入字符串所代表的十六进制值。

示例：

**var** number = aiman.toDecimal('0x15');

console.log(number); *// 21*

## aiman.fromDecimal

将一个数字，或者字符串形式的数字转为一个十六进制串。

参数：Number|String - 数字

返回值：String - 给定数字对应的十六进制表示。

示例：

**var** **value** = aiman.fromDecimal('21');

console.log(**value**); *// "0x15"*

## aiman.toBigNumber

将给定的数字或十六进制字符串转为BigNumber。

参数：

Number|String - 数字或十六进制格式的数字

返回值：

BigNumber - BigNumber的实例

示例：

**var** **value** = aiman.toBigNumber('200000000000000000000001');

console.log(**value**); *// instanceOf BigNumber*

console.log(**value**.toNumber()); *// 2.0000000000000002e+23*

console.log(**value**.toString(10)); *// '200000000000000000000001'*

# aiman.net

## aiman.net.listening

同步方式：aiman.net.listening

异步方式：aiman.net.getListener(callback(error, result){ ... })

此属性是只读的，表示当前连接的节点，是否正在listen网络连接与否。listen可以理解为接收。

返回值：

Boolean - true表示连接上的节点正在listen网络请求，否则返回false。

示例：

**var** listening = aiman.net.listening;

console.log("client listening: " + listening);

## aiman.net.peerCount

同步方式：aiman.net.peerCount

异步方式：aiman.net.getPeerCount(callback(error, result){ ... })

属性是只读的，返回连接节点已连上的其它节点的数量。

返回值：Number - 连接节点连上的其它节点的数量

示例：

**var** peerCount = aiman.net.peerCount;

console.log("Peer count: " + peerCount);

# aiman.man

包含MATRIX区块链相关的方法

示例：

**var** man = aiman.man;

## aiman.man.defaultBlock

使用下述方法时，会使用默认块设置，你也可以通过传入defaultBlock来覆盖默认配置。

aiman.man.getBalance()

aiman.man.getCode()

aiman.man.getTransactionCount()

aiman.man.getStorageAt()

aiman.man.call()

可选的块参数，可能下述值中的一个：

Number - 区块号

String - earliest，创世块。

String - latest，最近刚出的最新块，当前的区块头。

默认值是latest

返回值：

Number|String - 默认要查状态的区块号。

示例：

console.log("defaultBlock: " + aiman.man.defaultBlock);

aiman.man.defaultBlock = 231;

console.log("defaultBlock: " + aiman.man.defaultBlock);

## aiman.man.syncing

同步方式：aiman.man.syncing

异步方式：aiman.man.getSyncing(callback(error, result){ ... })

这个属性是只读的。如果正在同步，返回同步对象。否则返回false。

返回值：

Object|Boolean - 如果正在同步，返回含下面属性的同步对象。否则返回false。

返回值：

startingBlock：Number - 同步开始区块号

currentBlock: Number - 节点当前正在同步的区块号

highestBlock: Number - 预估要同步到的区块

**var** sync = aiman.man.syncing;

console.log(sync);

## aiman.man.coinbase

同步方式：aiman.man.coinbase

异步方式：aiman.man.getCoinbase(callback(error, result){ ... })

只读属性，节点配置的，如果挖矿成功奖励的地址。

返回值：String - 节点的挖矿奖励地址。

示例：

**var** coinbase = aiman.man.coinbase;

console.log(coinbase); *// "* 0x9749da31f3a451e182b8038df7c08f7863293ab6*"*

## aiman.man.mining

同步方式：aiman.man.mining

异步方式：aiman.man.getMining(callback(error, result){ ... })

属性只读，表示该节点是否配置挖矿。

返回值：

Boolean - true 表示配置挖矿，否则表示没有。

var mining = aiman.man.mining;

console.log(mining); // true **or** false

## aiman.man.hashrate

同步方式：aiman.man.hashrate

异步方式：aiman.man.getHashrate(callback(error, result){ ... })

属性只读，表示的是当前的每秒的哈希难度。

返回值：Number - 每秒的哈希数

示例：

**var** hashrate = aiman.man.hashrate;

console.log(hashrate);

## aiman.man.gasPrice

同步方式：aiman.man.gasPrice

异步方式：aiman.man.getGasPrice(callback(error, result){ ... })

属性是只读的，返回当前的gas价格。这个值由最近几个块的gas价格的中值[6](http://web3.tryblockchain.org/Web3.js-api-refrence.html#fn6)决定。

返回值：

BigNumber - 当前的gas价格的BigNumber实例，以zhu为单位。

**var** gasPrice = aiman.man.gasPrice;

console.log(gasPrice.toString(10)); *// "10000000000000"*

## aiman.man.accounts

同步方式：aiman.man.accounts

异步方式：aiman.man.getAccounts(callback(error, result){ ... })

只读属性，返回当前节点持有的帐户列表。

返回值：

Array - 节点持有的帐户列表。

示例：

**var** accounts = aiman.man.accounts;

console.log(accounts); // [ 'MAN.9HE223J2nC8HYjEBecdB1xGXFETG' ]

## aiman.man.blockNumber

同步方式：aiman.man.blockNumber

异步方式：aiman.man.getBlockNumber(callback(error, result){ ... })

属性只读，返回当前区块号。

返回值：

整数-当前区块号

**var** number = aiman.man.blockNumber;

console.log(number); *// 2744*

## aiman.man.getBalance

获得在指定区块时给定地址的余额。

参数：

String - 要查询余额的地址。

Number|String -（可选）如果不设置此值使用aiman.man.defaultBlock设定的块，否则使用指定的块。

Funciton - （可选）回调函数，用于支持异步的方式执行[async]。

返回值

OBJECT – 对应账户币种余额对象数组，结构如下：

* accountType：QUANTITY –整形，币种编号。0，表示主币种，MAN
* balance：QUANTITY –整形，余额，单位：zhu

示例代码

请求：

var result = aiman.man.getBalance(" MAN.21LDRQtTVBdv3EcktxrVQebpQmLEA", "latest");

返回：

[

{

"accountType": 0,

"balance": "0xaf572216ba03ec528459"

},

{

"accountType": 1,

"balance": "0x0"

},

{

"accountType": 2,

"balance": "0x0"

},

{

"accountType": 3,

"balance": "0x0"

},

{

"accountType": 4,

"balance": "0x0"

}

]

## aiman.man.getCode

获取指定地址的代码

参数：

String - 要获得代码的地址。

String  -字符串，币种名称

Number|String -（可选）如果未传递参数，默认使用aiman.man.defaultBlock定义的块，否则使用指定区块。

Function - 回调函数，用于支持异步的方式执行[async]。

返回值：String - 给定地址合约编译后的字节代码。

示例：

**var** code = aiman.man.getCode('MAN.4Pn182LSJ3JNr9by4T5kDKsf127Jb','MAN','latest')

console.log(code); *// 0x*

## aiman.man.getBlock

返回块号或区块哈希值所对应的区块

参数：

* Number|String -（可选）如果未传递参数，默认使用aiman.man.defaultBlock定义的块，否则使用指定区块。
* Boolean -（可选）默认值为false。true会将区块包含的所有交易作为对象返回。否则只返回交易的哈希。
* Function - 回调函数，用于支持异步的方式执行[async]。

返回值 - 区块对象：

* elect – 数组。选举信息
* hash – 字符串 区块的哈希
* miner - 字符串。这个区块获得奖励的矿工。
* mixHash – 字符串。混合哈希
* –对象
* nonce - 字符串，8字节。POW生成的哈希。当这个区块处于pending将会返回null。
* Number - 区块号。当这个区块处于pending将会返回null。
* parentHash - 字符串，32字节的父区块的哈希值。
* sha3Uncles - 字符串，32字节。叔区块的哈希值。
* sharding – 数组，分片信息
* signHash – 字符串，签名hash
* stateRoot – 数组，状态信息
* Version – 字符串。版本信息。
* VrfValue – 字符串。Vrf 信息

示例：

var info = aiman.man.getBlock(3150);

console.log(info);

/\*

{"VrfValue":"0x038b5d8384b06f4f6006f1e27803ea7d26fc4e2b43dff3ad6839334b784500be9d04fdae385d7a70c70ad8de0d1c35a8e6e974b0c851ebaed2ca7d44896692c4d5936bc8b5d7c3be2ef0376808faef3aa038b8149a54a5ca6dd53d47dc313e7f82b7718080a40ad323b5425664307f153c60845c7c62f6c6c5109983095faaa186823316e88ea050d736b911e680b93a434c90983be6ec2cd826bfd4777de60cab8e",

"difficulty":"690",

"elect":[],

"extraData":"0x",

"gasLimit":1050000000,

"gasUsed":0,

"hash":"0x199adb32a7a08a6f97df00b6c3ca5391a989e38ff6d0d5fd536e2b6b9a7fda4c",

"leader":"MAN.qTQXrAzCg6ZxPY1bEXW2Yfj8zB4d",

"miner":"MAN.4DnD4CZ1LNiHeb9SZNrn6XQfuETPC",

"mixHash":"0xe76f1e217cf57b29e928a8ab2cba8d62e76f1e217cf57b29e928a8ab2cba8d62",

"nettopology":{"Type":1,"NetTopologyData":null},

"nonce":"0x3a765fee9997be02",

"number":3150,

"parentHash":"0x83800c55e124eb51ddb88cc7760572e24a339d098554b033aacd5174e5088e47",

"sha3Uncles":"0x1dcc4de8dec75d7aab85b567b6ccd41ad312451b948a7413f0a142fd40d49347",

"sharding":[{"Root":"0xafaf9148ff954252a9a8aad63849f0e6daffe90b6a1708c23cff861a83e9eb55","Byte256":["0xe6590d918827229832b351eddefdd2ffa922852a72defc10e3eb12817e5a43ea","0x7163165f55280bee02522bb850dd807215447d570485757b6d2c556385ca056d","0xc5de1c1fe88658a98297cd9e0cc9a4fca32980c3eb097bd6b176b051c38ec5bb","0x21f0e0bfe74e06f0f368afc2f2ceb2b5642733c5ff71a702185eca2df63c3f97","0xdfb3d32d05462e4d7ac5e7fa0e4931d8eed705d8ae6a9d8899d936ad3dd9decd","0xe7d0455130d5b93fe2f5a987a5dd88d4a106fc5a0e58fe62f6e8b2cc117138cb","0xa4df5abd03a97ebf85d624bf50413c6361c5c4ab89e3330ddf6722963e6dc065","0x13bb3b2236a0c6c815316617b50fcd9d9163acb751fc5283913eaec04f9ae11e","0x75452404af28ae8eb6ee4d38446bf6a42b1a1165e9e250b46de35261d2864867","0x89deec31e14e282c0a5fc18144b00c79c31cea4754d4e589a7cf34550ca79b11","0x5dcd592532bec8f6382d0e3d305ba4a4b80aef1a311eaa2ebe44475a2af6c765","0x1ff0f9ca7c285ef174fe30c0e4048575687fff699f597f0e20ca3298e38efa5f","0x9fecfb9faedfdd7c46859f242f2c8f9e8d6314372386f478a90135fbf3952cb9","0x0391168fc774cec8df68f0c4e144319be13b3cf38788a2cdc6a828059515a14a","0x3605f2f12a3d2b3c3c92e12a0ba1ddeabdf20d4754b29a047af6ad6fbd49b39a","0x9b0be14f07e284ad130a2ab04422c109d56f763399cba695c63996daeb894f8e","0x70bd6c45c0ed54bfc4f61dee1a25782cfd4ac5a9ae068de54427e7f1e399bbfc","0x971e9cdf52213bb549ce26e80dc94dd4efac812ec0eb786723505f3b471d5d4d","0x699532b14721f9fd71b458b4c52fdbe55c2c848f1333a80274e281ef36dd526d","0x234a1cfe34a17439982aa6d9ec7531948436fbfe8f1415f0c52130e78ef457f1","0xb3fc99671bf1d4167aed47e533cf57e810babbe83de9d9b30d63ad64f587deac","0x9a37434dfcc65fa92aaacf8968e7f11dbe91f68417761d231e83d5fcadb31cc9","0xf763a80df775939c25446e1abbd817f85a72bbc234193dd6de3ddb9300dee39b","0x20dcaeaf049fa153c796f82c392deeaf79bbfb1eb16c7bdec15e6cefab9ea3a5","0x72ac1a4d8db6e6775dc68ac34ac54a6247c672d71c3350cb92ac2fd718f208e5","0xd0c8382fefacb87cfa0bb0e50fae89c090cce0c40502d1148a9c53087d04fb87","0x52719a079d3740b52778497173ca3bebbfeb6009e6af2a7055e816631689c7c0","0x40e8495a046c71f6791975c7ed29bbda697b871ecb6aae3222d776892734b8d1","0xb5c5bb6e73cffc4108b77beb3b417f253c5901c679b97ece6bed6eb88ec2237b","0x0416a2871f6ea7ad8a95d2365ef77c72318fc47084b2714d6245d456978f1a54","0xbe7a609193640832f1cbeb9af1162ff7179b0b3516b5aa79b33f1b71f85f95e6","0xd0cd185766c029a32af30cd825cede93c5fbae14bb065c8ece2bcb2eb70e702b","0x288e21626e217c39884555e6c5b8bffac62d6bb06084efae613bb4970a7f7452","0x3392e641ffc12f7eeee881114a9258654d9b331d8ec8aaffd04241c7e541ccde","0xce0e6f0872e991c1836f1ad299982f15c9881cb74960339c67b01081101df1e0","0x9f73c49264ebba52f8ddf0c172c93ce27f8a7b493cef701287f6fe63ad22550a","0xf7dcc5579e4104225e7231b18646d9c376319d67299ee9aaa432e341a4d26116","0x4d3cf0e7795b7763b40e8232a458b3653d40a13ce91a676444bdc20242fb1ee1","0xc92f0f6d4d1f9644ab367fab98187c3a334a833510dc2b32e85155bd7ef21312","0x756fe107b7ec9ad076c1c5ca6c0bc18060446e56ce5b50bb398c2331b88c6b4d","0x547dd89a5e22b122b3ffcba4b155a7979e8ef4dcb81c8271a86a4a1e640e629e","0x3a2e34b40a053dc5c4559a55f583c995d406ac0d53f50fb39c20f1d847c3adbf","0x1b6448957b61c7ac9d2c0f74b3bfaddc41108fc6545ba4dc90a549b3ba834a76","0x94d341f9d0384ed1898acd2da4a53d10e21e9521ead4b31a5ad8e72fd9f32c7b","0x9539b39dfa6d9e7af40529a20f056e28333beca05ec16b95ae5184ffac96cf2e","0x7a4279d69ef2c58cf0f14325c8e1b929b7117a08f184ecd1a87c17973f8cc7c7","0x9afe12f36ff66e669a1ba8cc5b14fb7070d7fa34036023a9916c0624a15c2931","0xa868ef9c103c004ce52b175385df3621d737f40a3d4931b68a0da8ed26d2bddc","0xa0334b29d8cd5ef96f2c086bdc4fde93705c0f81397e5e6c81169c7ec23f1d8c","0x13123b3734537e98843a49087dc8cb07ef1adda95383da846a33fecfae2e6c92","0xb105262159e3a0a5a5b776d32331f713c7e9abfeb19731d8dc412713c83884e3","0x90b512b0d8aa7737b7cf8b346bbd424b44030b627a6e0799dd4fb2cdcc1879a2","0xa5bf0695bccdbd4c24ad7ca5d716f758a1e450cce380fe6cb6bad44f636b9e77","0xa648f02943fa9ed8afe48779ad82edeb13769eacc332ed99bb0df7adb39a331d","0x1da012faf7f0c94d382971c16b65839b4e7ee58028579a775a146d8322c2daef","0x9e14356f94e6445f4bddde2ba62feb8af547aebc40957cc78a0615eac38c04ab","0xf201ec9dd3d6f80963bfdee0af2e939f380b33919db27d03029b32b12918e397","0x61781aa36cae99cba90265d70a49060df19a139397a182f39001dc91fe0cea45","0x4ba711017308c0c00d9601b47560d21fe9d698bbdfb4dcfe4e41399d579de477","0x351711e48e82840efa0e60197c7679e50116eae728c881c9848da08e76189a03","0x8aada78a5223d38d7a5388c9084c8bdf2d37033e9afa6104ec8061be062afa1e","0x8103a8ab498e48a36aaa4bbac38e5abd56aac8a57cf25b13f942c8c5a3dc5374","0x769d9eb87946d830d00000aa9a5d83b9931416f80fff5ff4925c7d3414c81df6","0x9f1a67bd31f4f87e62cef8491016b718d2c64588da25e1da0ff0d6de461070a2","0x40b2be055265b0117d45869b0902ee23bb9bf6ae59ac0886a41b2b40f7a928c5","0x7fe06c8b32387b553992666a30d00e835b5aa3b8441054cc78734ed0d173a24e","0x5c34557e038bd7b1754dc02605956b546b4a72fe6cebb200d0f2403c42a77880","0xbf29a6a4985f8471b5015dc222475b5d9710c25852746f24deb4507cd0ddffb0","0x1638721471fb466150dd5e32eaa9cdf6dcb62ba4fbf4f844e1385b8b5e02dd3c","0xef238acd785955cd06295bec91f445fa16ad3a4ad695082cc75ba8a94217cb80","0x413ba7fa12dd7b3770e7766b40d233541432838ad17254654e7d99ff214a9dcf","0xb8f3ba91bb941cbdc9e41ae3c14c8ff419633b3e59cfa063e4dd01185a6d9c08","0x62b9373fa92a8049ceaea4b5e3ef218711e30da252e7133d1e44f4da1a626d69","0x0c6116f9d499db8dedb54c906bfb9175094989bedd619f519b18ef1cf966d412","0x14834a646d74c07c89ff67136523b822477ba1172bc503297bd967ec7d8ece24","0xd8c574fd52984134aeb490615ee08905447c219dae53946bf465dcc90546a93e","0x28ee89c2fb1611273459e162fd7663518d46b406eefe41a881f243ec74888b8c","0x721fb8f5b55afb4c8708f77824ae3bf2f12937465396b97a2a80739c2b786366","0x2e39ae5f7776a121fd73ca32a534954332eea5bd55294abd2f6af382fba7eaa9","0xbacc4c0db2ca52c35deb0e14cb6f3a721491f2651915ca1ea225356108eaf115","0x0f8205ffce5451a62394babf1f448e96a8a113fec01549f52a52065e57a9cac6","0x678d574571323db93ff07e87b5e1530bd823446cfe68eb1b095bd4e2048c19d6","0xb47c22e3ebb1c72a293210bea5c3662905f41d40b8f035066738c47c346208de","0x38085f62a6f684ce0ac34d76cbab0b0beb0098cd186fefa898d23ce9ff905f7b","0xdabfd9bf5d6adfe6df389164c5782b9b0baf48efe6ff29e80886e74ce07a40e5","0x4175bca51277ff969e5513d34cf3ea5cb12128fb0ffb42eb74b0ea922e7d6b43","0xff6b6d383f7863f3cfdc9f55772c80bd8c90ca3fe7b757abc943720f7e358a76","0xae3063b849fd8c43a8fe147cd4c4621f211572622f8a94f88510ff276d001baf","0xa6c88bbe6c083324c88f725f58b9ec6ae9720a2943e8ab32d76ec45cc1a6b9e0","0x91d78cc6349e4a00c485b545b6775fbf373a497d03f4fb7c5f630252387f532a","0x02dd65b990c9221ed78f5a9cf40524d72192a83afdd2ba717774b873c557b5f4","0x137c3716409470f5db914c4ba0565dd61a0e0ff32f36620ef9cf02801a32dafe","0xff57c9d79d649a463adfdb1ef7df84d92cef0605c62998060bb5c4adf59ac424","0x3fbb48e0094c823de6c7b4e223981fa7bda805e36f58773e86eef8f86d669c5c","0x04bf865d6eee647795a2bf5aecb34d61cdf153b6271df5cbb4fffa8981cbbdd9","0x6ad3357f88240bac475711f9cff875cbb2cda070b74331ba366cfa629fa8e93d","0x6cdec23a2e1f529a86e756bae00b43d859ded2d76112a1f6855a898f8d8b490b","0x4ff46618a6825eaea9eab5430d57fdf6625fc6278d7df1a4d6a0d19ec7e219bb","0x4242fd03771263ec45b36dc92db602d545c89ecdc5e5b7b66521319cef1a3d1f","0xd3198dfc98f890cb856446b69e2765c33c63258ca608fb066686d3b699929343","0x020923cf821d1e31f6cad317e093ca147b85180379d482bb4445b19b127edb5c","0x52fcd2bd2b4db91cac6a4867920275540cc28de87eefdb8396bba69157db004d","0xb47d1fd69b8645e12d1f8ff6aa1b1537ee5441678f82408e2d6e4a02fed5b2ab","0x0256a37db4d6c325b507ed801ff18d1da0275940e190b31cd8aca1f567bc3f8e","0xaa6baf2af1428ee2ef818768b435cb4572d34907169dd196dabba01020664828","0x08a9fa6c9aecfe5e41238148487236a9578547eb78639bf3db67d1ef10a4066e","0x40e7e542bdd72c8342294713cdebc4a0283ed2f14dc70ae14a5f230acf73c249","0xf8d379f256cd7c3a9c2332cbf0ff5ac631f6cfcb7e675dfac2dae8fc72c5cf22","0x297f0134889eef56a64216b584394d49468343a5834499ba52bd824b04960fae","0x67c1328f094a914bd5f9568d8ea57d99d9af0b8dc1f6450e3cab95ce0628424e","0xbff1a8a2b96c6e8156332bf4c3b57f06aad869a6dca895cb2b14ddd58706649a","0xa5d4c497a6ce8681c9d547bef6451c2016686dded565cde71429a481757e4234","0xf57764735a1ee377ce639674d1a07b0f6cdebf6b0c44c61dca646fc11c5f1f6f","0xbf3be5cf6becd367fbc6313812ee60349f5ff8b60854cf7224289cf07dea768f","0x8b7a74a6eee5ffd21d43fde7cd1d8e5dce0d698bcfb5a64c8cc455651cdbbf66","0x4cff2483bdc053c9c10f257ac6aabc5d47d5875cb132be1b9ebb378029eab988","0xfefcec28b72dfa27152759c5bc9969267b40d88588741d34523b843471e1881d","0x291ad9735c7265151d280025c65ccb411db71513b1e2bf9573ad75ec36b76d0f","0x22a0cec16cb2c7acc95f4f9eba7ab0c4fa999a56828260cf0ac52849b7736a36","0x20120404b96c4d378f8419c78f626d1b8b8b4b12311de3dd559b9fedae3175ff","0x9131cdaa15c62d798e93c52eea9e50a9fe71a396da926891318f843747306b69","0xfe9a9a9db3f84f77c7a21ec3badac9e4834502e7cb680e466dea637925a6a566","0x21c4c2d904141ad3352e221f23fb05d61a521c5082a4dfd6f179ddcc3c591531","0x666188ee8e31db7c5ea44d2b0e01f01ddcc299082d0f593fb7d17e4b300efd28","0x110b0dfee5de3e846afb6e7abd120f6b07c71193705471bdf2afccf2b3ec6e27","0xf4bb37a12f26eef416c1e8c1d7e67d2bc2a0c3da627bdd79fca5d62abad6ea5d","0x5e63854709ac743dbb840bfbe034e040d22235e58a6b74186c715a37a42c2201","0x3bef4403866b7604d50d132e5a7e13ca2b6dcd2df93a4e23d285fe76c9d36347","0x3358308885d63dd54a50336f30ee433f5aa980b31b0a6c5bc8e4cb96df410255","0x55f70387c523df74e9d20a77a67cbde75b6f153ac12ade27383fae797c9a5ac3","0xa7df48178569dde629880bee5e44f1ded1374a89f09e436ba34f2829f8d4caa7","0x1c7a5e557b4af373f8d6c183748200d7fcfd09d7142b4c20a3842fd585fb5707","0xc770b41054624cc864daec7f250ee8fe781ba6e0e2e0d8e9610597439e3e06da","0xf92cb15120b13a95bf4de3c5c8e4b14db7d7eda9b877378212e0cdd57dc4dc90","0x5f77df2a0e5b8cf424f45f9f9978a285c7711e61773aaa3573cfca6bd54f0a41","0x7504b29f0c65ab498a9a1ccad5866b96daa35189513895d5ea06f54f2351c52b","0x06ac11e45ceb7a53aa8dd5104c21aab558e7c64caff3f5d587d8aa04790d9822","0x2e288ed7289094e87ca9d2c4c6c6200499b8eeb0b79638bec71cb9df0af5a489","0xe1d2fbf3bbbe25e9a1202910ee6907326058e5dc9891151a35e2e4e449203d1d","0x7a055ed069ee0ada54155b6c6651ae8a615ad057b5633876f3dd9d0c43d97729","0xc3b54cecd3f14501bf9e8593099fb3859e04f6768f2e54600df55949f22a0037","0x6cb2115b57d9d315d4fe16c385c2745fe33ed8882bff2ef302cca2d716f9908a","0x16258ee59baf5ffc2274cbf0ea137c2c15b5f2aafc69de30dc279985f58c48ce","0xcba12563131731a767d300db3e75a3f6dae8f55d252e5032bc625719efb2436e","0x23e465f11c1050b30385ef99c16346c1cfb430105512c77d914e369523e7b3c4","0x7e7de9683d90899a25d50cee103269c143435bf792caff5a1bbf51a654cafa4e","0x16677d9766a3889603a9f5aaf1e20e7cc8547abae277a8ac32904166df11716d","0xbe64d258666f5295b77fe67a778c844aca02cd4fcebde871ec48666ec35e2753","0xbb29933e3f0cfa4d08ad53eae5701e7a922053f5f7ff9622decfe7dfcc3cbfb9","0xb1fe88e10018afbda76a9cf019db7d5181d54a0aabc759b381409eb058974234","0xfc48952e1266841d9d5a569f51bde55eeb6e180398b5889d573e2267040c7fdf","0x2a48736588da2cc1ed175507f40b6058a307d8a481c92cc93ab8d6700bff1927","0x90e376a96b6a8391fcd2fef254f0baca9bcd6d0670fb733689edb1c83b1d62db","0x6559999cdc81ff7bd75c6b8953c7a18174f03f3b0a92bdcf324f38a34b35a143","0x2df96271ef834a40278ed1a41610cf7ff4946b7783e0e16c5e585c0f2ad583c5","0x95ffd0dbf463c2c5feac86c6f92f9e6655c612fa30bc3c41c63b62cfba2b0f73","0x1b4d8dfde74b8d479c59a24516ec887bf1b62a8bc5a70b7b7d43b312991a0806","0xe01db0fcf41e6e89d4fc056b3b71314f1a10daa28d0433d175620c2c2dca5cd9","0xe7b2faacb699911eeac61056bd18a8efa2453fedace0035bfb0025ab20e40a6f","0x31a1e84826638b6f89583fef0b5ff904e0277bfbe493a594c1b496d6543c6b90","0xf16992183a0bc4695a46e7ca77d59d091582dac4c9e04cca361886dc2fb8e276","0x9cfbf475cc1da24b8bb37a8db378efbd6e8b08191388c1fc0ae5c806a142eaed","0xadb2378580665af002b263d304e7300e776c3e90f6b82bfa4dfc2101911def68","0x3790e1d138725c247b0295ae4321b6be0a241e82bc70884fd1baaab1825df513","0x5df8dcab4b83d585fc87b240840cd455319a8173fb3a72715da0609fb2e8a030","0xfbbece8ddec2aa0e23972775b47b560c2b7f9cd793198daed2818f1a219de44c","0x206cf5c26897225a482017d4538919ee068bdeab6373759a65f087b479691174","0x027010b9a36ad57664a5fd91b5bd7719622dedba1916f2365ff56b1d79b2762a","0xf49573c17a156fb10d5713ac19d05622a4d4087c2a2725c49c9715cb31cd6070","0x18e618d99eb1e5d2ec654f728925cd5e1db7393c274f45885826e1feaf4987b6","0x7ddedcb75a608a5839b8903b49ba0c658a6da9a0bff98b34d540fbe5b41ce62f","0x92072e87df3c6f2f0ad57b023678353dfe209eaf8b8d4369cdec5f033df2dda3","0x9a3f62f63c7e870b63bc80b8ef49334bafdac68270546ccfdd2ebcf1ff0c8e44","0x87598abd8196418d8ab34f73c49e78a2929bb974f7858858348c276101afc7a2","0x78a04c6b2dddda8424a466e7aeb4898798e111686cae3c23017391d84407d140","0x1d8b61f91164ce93eae82f12d89d2530785845ebb3093de6ead31d148b6b1fb8","0xb01932d0603d0065e5d23a1737f6b28bf985d89e32d4be5b5f54c688e4346e8c","0xdcbbe27b67e33190b637ace4f33d1738451d0653a5ae4dab2cce7cf7ed4e42ae","0x7377f3ff5f30daa3d8a1aa4a1009f9749a769fa7eedfd5728845310221ffed62","0x2798517065c1e3b3d3857e990fddc0407412bb2c92cbda1b2575faa6792e5042","0x645e852af5d450f1b215f95cb06153640e32ce94811e553465eb0d004b869c08","0xb87687d539d4c1911375b4f4127e4fd7d2cf4827431b4e0c481298d4157c82cd","0xcae9920f196fb11f5625a8dd6ea9449c3882d833f269a4781f1b6c7201464d39","0x161bf9474bc762aa131b9449bc556341f635d2bafd685d3d82d28383759397d9","0xcec323cbd363c288259ace37d22b9c9361cdc535a8a7735e70f69670ab191e21","0x9fde3d2c1f4f53a58f7098f590badf9be377ce6ac60d94ccc4641a90a88eb7ad","0xb278c66a18f16e8a16987608e8c5e4f06f2612bb6e22f290f67a614f96f6512a","0x990aa6ab928bad88fddd243ff5fef79bb2d07f3472515aeebb9eaadfee79c54b","0xe6b6cdc7bdde0798bb76c590e480c19b434b988b657d6b34e7a233ca42ff0197","0x122dcb8ef90f4ce69e3e9a453ee4ec975f22ef41fe53163595d1e1610e79e075","0xa7c6819f9ab6b144b762273941c47beaac208b433fcff4f8f31d84ef558cb061","0xd64e213b13ad576e6ac19ccd7de2b511f398477d9fe3443f7cf2a29005de06bc","0x43c9cbf5a38b11f7f54f8f7c4cdae2857372a8700192b038b36dbf139d075ff3","0x0f4f7492ca85edc4b9cbb51b3bb2a4ff813dbc61f16bf3f72f4bad39392d23a6","0x6932384e0ce5e88fff48e97d1d388b55100d1c932817b421ee8fd904fe22df47","0x5743b18392ff55c3b4116cdc2d94f16979ba0728e6f94e1cc3506edf8176c7f3","0x0bb9ac5e5c6ab4badce637c08775d01b139c5d185e71184d6b3e919960522212","0x359d87f6216074504e6efb22d96b932417aabe85063146d840d4f98fe875f97c","0x55cd314f9b6e179164ed3363d845c6a57a6f9f266dc9f93f78a20a5302985811","0xfaee4bbb7391caaf9bc68e4c2d0a4b401b78b0a0c4f2d79c8420bf6fce006dee","0xf7caae8fa5f18012aa9251bd2713a1b96524ab35ddded12c1486b0d37049b202","0x1b331782c4f6e9bd5df7c63e46ffae72d80e1798559ee9c9e49855c39a0ea096","0x7340fc29d0b716fea37015088a5608da394553eca1a12d33bb63ddb3304497dc","0x6bb5b4e55025a5e106b492112828688e7e8b2a2ad4e1ea72918da75e88271582","0x86c8c1bb9d684b5885ac1ee734b0aa8faed2229eb12ad382993a031eab607747","0x620c755cc27d355105ed7d66014a125999cea6db4afe072549438001ecd4757c","0x8ba7e585ff17c458cbf6a40db234f3b5859fa529a6b5f6609b32ca8206256fe7","0xac5d6dcc299a5713cb741e50bac4073efe5fc8303b901194300df46f35edcc60","0x16007207f7ccede26f9b9523e5dbcbafae53ce1bccfe1449e9b9cac7906ba850","0x98093f2773810eb201586bd6e0b112cff53d74707cc61ffa250dcff6baedc198","0xfd8baa0b48768ed0280db6497bf8017b03f6a0f6584469fb4fb764b616d2ddc5","0x8784f0b933ba5178be11298b1b75ff604288542605cf65c7840606ac526b1517","0xd00f3d92b364de15728e90ee271df3713affbc3dc70be0d2d3f46a81ffa46da9","0x71f2294972442308afd85b00411428013e1f827b845bead91caffd383e212e79","0x24e595289faee4a1809ffb9bc8d782ed0b414144ea618fbef8775231ddec07ad","0xa1cd74cf63003cb527551b6566c1bba5a072a46b0bb5db3ee1dfa8cc08bab9dd","0xd24c91e40d83b034b6d401c898a01fcf06c9d6514a7efc77b3d8ac3a7f12648f","0x1fe3380ca9600854fa0c09cd468c782343930bd17b8388f3cbfd7c9761b568c4","0x3f8b5a828cecdf4127ccaf44332919843a825f9e6d01662e0a0bb4d76719390f","0x3ff37816f7fbeaeafe34a7c3721185371646660869d272bc7eb69da7f6673fd0","0xbc11e08c0d4ee9a349d53a504abbebd965c0f4d0e4e9ba7eced6f0a264a6932c","0x194b843e93eff7fb6bbccb727c77cc097633f9bfa84f2ad217fceb638367e7fd","0x07795ec819e52a82b9a262f1d95c4dc6c337db7d94f7aaa7a7992da901347467","0x1ff792174ab21acbfeec63d0a4b2efb90cab7187fb0a6df502ec868131b09038","0xf06584200d16fdc2a275aa7835d57ccc7ea8dcad877e09a36db125f4a0b94a4b","0xa72112a00effb680fa719f70b3304f877e04801974d270e3c074f8102dbe606e","0xca8821bf566a12a556a35476f21a5a4a7c706a0038f46365cbf644efb0dee8f8","0x3ddb95460f645f90203f85f338d760963e0ee04c4efb2f9dd72d25c73b27f705","0xabbfe4f928b4846bb37d0e84ae1796f1f736d8d317ed7c91f94ec7dcadc59dce","0xb4e55742b56f2d51261fb803729a2a679fce12fb3ecbaf8b2500b0af32cf14ad","0xa966be7f0db63633decd03d9e517f589fed37401656554ed7e0a00a21c5aeda9","0x643e12c631f2404ef63802109d1d8593d4d25699e2fc385a08322f873680a2ff","0x3c935be7fe5d8ffe4a47a0ff58985c080c39156900e119981ef833d6519afed1","0xbdbb6c4bc339a45efdb79e4cb3172fd4e47d47e47236b7ab636786b3c6e4d82c","0x55ba55d3fa40491e03b70d03aa4c5139c210726130783672f2c7b55f4ec5500b","0x3e0eff0faeaccf64d6b306ff073e5472947ce045ac97c2f547730a6f56fcdc4c","0xaa41263a0957d92c3fbc9a0e6d260c14a3e65c0c1a37ddb044dc6d00481a307c","0x8cdb7c42597116494b402e7badfe0e9440588c1b6c997a66d3ccfe4d12bab6ed","0x418ae305af89e63feaf0e1a16b33016729855f2214044d1a293a6fb7eff8d22b","0x690027f8ab5fbd4b770d4215bee4fd0b3b27a0143af1d9ec5f11859e7b72ab8f","0x2554e968489ed202ee17f266e518947acd02647fed1017baec68e285ea794ce8","0xce18a019b280c741c9969fc127dddd32c2be450daa3f96dfd15e8443e51206e0","0x7e19d69551f6e4a0bd747386a3256c715e9c0fe42b50734b0f67864e4a4b1574","0x73c51fca8f96dfd5640c3ae1fa9e8a195d94d5d58bb5b88b63aa7caa7fdac09f","0x4ee60e6af2c56f4e9f2e2ec4c1cb933b60269fee57f26fe6437b279c944ffdbc","0xca81ab22c32c996588cb0fa6dcdb8867d3e38becc629e7714a07f1d16a123da6","0xa5b08976e4c7c07ec3eb96003e7559d4d9c569736f5e3a5059a7768c778a845b","0xca29176c624becd00e3568390a89cf079f21088248f7d0a3db7855fb383cb52c","0xd4455c24de8ae99406e93d697b2c6a91cf0c8499b364c55e3101a857b123ef99","0x83dd11f3983cd5580689f0c785b71b7743c7f79e8717fc6ccfeb3217c88379da","0xbcbdbd3fe90375cc470e61ff14a2c872c8a1ba586bc9e4fe0120b5878e884caa","0xe59c7749d113c120e8c85cb857a684308013fe581dac6db3c73f76af9fee9003","0x34fa84718e245f11efc2f5fa059741886cf789adbe9aaa7f578bf049f890bd5e","0xc30e67a73ff4aec5a2f8ca4d76302009f551108ef5602d064f486be499bd4cef","0x5d9613b6e0ee24b4413fb2ec986859f57ba3272dfc4cf7d6939d4ec2e925c636","0x3d9af48134b80914da76d7a225d6f31c5b4215b3dd6d46621c828b17a46cac38"]}],

"signHash":"0x759f9e22b215519de1085404835f17f5d828633fc7e811040e17e48c99b74e2d",

"signatures":["0x7027759d43fd87a25383c4dc7c255eafaba8b9797ea91c215795a71301999b9f7cf0ec7793a156014ec7423009ad12a87fcdd3fa0c8170bd1678a92a4e27cd6e01","0xd994d982501b04469dc71442b623063a8af943c1fde289503396910eca5af042087f2229f89e9e331fff76b55fcf92ed34ede513199824496db752f18dfb236901","0x771b31760dbee7ff15f6443ddb9b8df95cd277b09da097ac0a34a5e1e6f5a2f322a7bdf72781e55fcf35e456aa7de6d55bbc463f7c7fdcf1022aa463d8682dbe01","0xd03cdd43823d7f210ee12d5804326956cdc70afb8ef0732e262c2aab164e93562f4ae7a8e517d119cb2d3e388de4178f1a22d6601acc72efec538f34c3658ad901","0xe47652bf5fff54158cf0ab32498375980189de184b1580393b6c2b9db8a775d84d63d027ff8aae0da6fb75dcd0dee442cea886a93d3cb0c0d557b1e928ccd03901","0x87d094568f51c56c40ba1b45131af0b72d27cde3922e0cf209aa258b07611a860b89ddd44a587ada7ce39a456bfb81cd9a2ed69ae3ac6dd4f82c6a852cc824fa00","0xe216381b5cce085a8922a930ebcf741111bb19d4adbf06eece9084d9ca78c6fb4a590077cb335c4a82bc41da74c69e31edb82e61ba80957e31f609b3c4fefab100"],

"size":10420,

"stateRoot":[{"Cointyp":"MAN","Root":"0xafaf9148ff954252a9a8aad63849f0e6daffe90b6a1708c23cff861a83e9eb55","TxHash":"0x91043e03d3576dd34f536c31d649d1149b43220d04757e9341d03d108b7b1393","ReceiptHash":"0x34da222b8545ec839225d66f32c6b29eee304c1e417bfe8bb9e0fcbc834fe148","Bloom":[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]}],

"timestamp":1562260538,

"totalDifficulty":"2862609",

"transactions":{"MAN":["0x459b9c910d46527befe5ccfe41a21b30ea9056a118e1ee86662b189ad9601556","0x6a0f6e9984a33a084e1cc3cfafed527b82962aef4ab590500febfd6de6107088"]},

"uncles":[],

"version":"0x312e302e302e33"}

\*/

## aiman.man.getBlockTransactionCount

返回指定区块的交易数量。

参数：

* Number|String -使用指定区块。
* Function - 回调函数，用于支持异步的方式执行[async]。

返回值：Nubmer - 给定区块的交易数量。

示例：

**var** number = aiman.man.getBlockTransactionCount(122);

console.log(number); *//*

## aiman.man.getTransaction

aiman.man.getTransaction(transactionHash [, callback])

返回匹配指定交易哈希值的交易。

参数：

* String - 交易的哈希值。
* Function - 回调函数，用于支持异步的方式执行[async]。

返回值：Object - 一个交易对象

* hash: String - 32字节，交易的哈希值。
* nonce: Number - 交易的发起者在之前进行过的交易数量。
* blockHash: String - 32字节。交易所在区块的哈希值。当这个区块处于pending将会返回null。
* blockNumber: Number - 交易所在区块的块号。当这个区块处于pending将会返回null。
* transactionIndex: Number - 整数。交易在区块中的序号。当这个区块处于pending将会返回null。
* from: String - 20字节，交易发起者的地址。
* to: String - 20字节，交易接收者的地址。当这个区块处于pending将会返回null。
* value: BigNumber - 交易附带的货币量，单位为zhu。
* gasPrice: BigNumber - 交易发起者配置的gas价格，单位是zhu。
* gas: Number - 交易发起者提供的gas。.
* input: String - 交易附带的数据。
* IsEntrustTx – 是否为授权交易
* Currency – 币种名称
* CommitTime – 交易上链时间
* matrixType – 交易类型
* extra\_to – 交易扩展，支持一对多交易。
* TxEnterType – 整数，入池类型
* v – 字符串，
* r – 字符串
* s – 字符串

示例：

var **transaction** = aiman.man.getTransaction('0x8838fd149ea46f4c182f2de9b465b731d6cb67794233602dddb5362474122841');

console.log(**transaction**);

/\*

{"blockHash":"0x1d7d46eeee21cfb1028bc85d254e0cd432d3cc128624fc219340e0ee93db8f65",

"blockNumber":216803,

"from":"MAN.2nRsUetjWAaYUizRkgBxGETimfUTz",

"gas":0,

"gasPrice":"0",

"hash":"0x9ad5310bf83afcbd2f3fb2b7da43f03cc1b9ae45af15d706541c7d17bbaa333e",

"input":"0x",

"nonce":4503599627799762,

"to":"MAN.5xYzBHrJfXeJi9yQ8Qq8hvm19bU4",

"transactionIndex":0,

"value":"600000000000000000",

"v":"0x0",

"r":"0x0",

"s":"0x0",、"TxEnterType":0,

"IsEntrustTx":false,

"Currency":"MAN",

"CommitTime":"0x0",

"matrixType":2,

"extra\_to":[{"to":"MAN.YaMnyEHkBFsx9mMKweDFv6qc6PCc","value":"0x853a0d2313c0000","input":"0x"},{"to":"MAN.2J6nGbdsqu8XSMfWWDKbYDSBmYFSF","value":"0x6a94d74f4300000","input":"0x"},{"to":"MAN.2M2AG1FspqrbBfWNuA3Hzd6DJsFdr","value":"0x853a0d2313c0000","input":"0x"},{"to":"MAN.2WeBpo7BxfUxVmryJrqLSAKwxMW2U","value":"0x22f8d6a601fc0000","input":"0x"}]}

/

## aiman.man.getTransactionFromBlock

getTransactionFromBlock(hashStringOrNumber, indexNumber,Currency,[, callback])

返回指定区块的指定序号的交易。

参数：

* String - 区块号或哈希。或者是earliest，latest。查看aiman.man.defaultBlock了解可选值。
* Number - 交易的序号。
* Currency- 币种名称
* Function - 回调函数，用于支持异步的方式执行[async]。

返回值：

Object - 交易对象，详见aiman.man.getTransaction

示例：

**var** transaction = aiman.man.getTransactionFromBlock('0x4534534534', 2);

console.log(transaction); *// see aiman.man.getTransaction*

## aiman.man.getTransactionReceipt

aiman.man.getTransactionReceipt(hashString [, callback])

通过一个交易哈希，返回一个交易的收据。

备注：处于pending状态的交易，收据是不可用的。

参数：

* String - 交易的哈希
* Function - 回调函数，用于支持异步的方式执行[async]。

返回值：Object - 交易的收据对象，如果找不到返回null

* blockHash: String - 32字节，这个交易所在区块的哈希。
* blockNumber: Number - 交易所在区块的块号。
* transactionHash: String - 32字节，交易的哈希值。
* transactionIndex: Number - 交易在区块里面的序号，整数。
* from: String -交易发送者的地址。
* to: String -交易接收者的地址。如果是一个合约创建的交易，返回null。
* cumulativeGasUsed: Number - 当前交易执行后累计花费的gas总值[10](http://web3.tryblockchain.org/Web3.js-api-refrence.html#fn10)。
* gasUsed: Number - 执行当前这个交易单独花费的gas。
* contractAddress: String -创建的合约地址。如果是一个合约创建交易，返回合约地址，其它情况返回null。
* logs: Array - 这个交易产生的日志对象数组。

示例：

**var** receipt = aiman.man.getTransactionReceipt('0x9fc76417374aa880d4449a1f7f31ec597f00b1f6f3dd2d66f4c9c6c445836d8b');

console.log(receipt);

{

"transactionHash": "0x9fc76417374aa880d4449a1f7f31ec597f00b1f6f3dd2d66f4c9c6c445836d8b",

"transactionIndex": 0,

"blockHash": "0xef95f2f1ed3ca60b048b4bf67cde2195961e0bba6f70bcbea9a2c4e133e34b46",

"blockNumber": 3,

"contractAddress": "MAN.38nGzwi5Xn5ApxHXquT8ALaMLpbyG",

"cumulativeGasUsed": 314159,

"gasUsed": 30234,

"logs": [{

*// logs as returned by getFilterLogs, etc.*

}, ...]

}

## aiman.man.getTransactionCount

aiman.man.getTransactionCount(addressString[ , defaultBlock]， [, callback])

返回指定地址发起的交易数。

参数：

String - 要获得交易数的地址。

Number|String -（可选）如果未传递参数，默认使用aiman.man.defaultBlock定义的块，否则使用指定区块。

Function - 回调函数，用于支持异步的方式执行[async]。

返回值：

Number - 指定地址发送的交易数量。 起始值为 0x10000000000000//十进制4503599627370496

示例：

**var** number = aiman.man.getTransactionCount("MAN.38nGzwi5Xn5ApxHXquT8ALaMLpbyG");

console.log(number); *//*

## aiman.man.sendTransaction

aiman.man.sendTransaction(transactionObject [, callback])

发送一个交易到网络。

参数：

Object - 要发送的交易对象。

* from: String - 指定的发送者的地址。如果不指定，使用aiman.man.defaultAccount。
* to: String - （可选）交易消息的目标地址，如果是合约创建，则不选.
* value: Number|String|BigNumber - （可选）交易携带的货币量，以zhu为单位。如果合约创建交易，则为初始的基金。
* gas: Number|String|BigNumber - （可选）默认是自动，交易可使用的gas，未使用的gas会退回。
* gasPrice: Number|String|BigNumber - （可选）默认是自动确定，交易的gas价格，默认是网络gas价格的平均值 。
* data: String - （可选）或者包含相关数据的字节字符串，如果是合约创建，则是初始化要用到的代码。
* nonce: Number - （可选）整数，使用此值，可以允许你覆盖你自己的相同nonce的，正在pending中的交易。
* Function - 回调函数，用于支持异步的方式执行[async]。
* Currency – 币种名称

返回值：

String - 32字节的交易哈希串。用16进制表示。

如果交易是一个合约创建，请使用aiman.man.getTransactionReceipt()在交易完成后获取合约的地址。

示例：

*// compiled solidity source code using https://chrisman.github.io/cpp-manereum/*

**var** code = "603d80600c6000396000f3007c01000000000000000000000000000000000000000000000000000000006000350463c6888fa18114602d57005b6007600435028060005260206000f3";

aiman.man.sendTransaction({data: code}, **function**(err, address) {

**if** (!err)

console.log(address); *// "*MAN.38nGzwi5Xn5ApxHXquT8ALaMLpbyG*"*

});

## aiman.man.sendRawTransaction

aiman.man.sendRawTransaction(signedTransactionJSONObject [, callback])

发送一个已经签名的交易。

参数：

* Object- 签名交易JSON对象。
* Function - 回调函数，用于支持异步的方式执行[async]。

返回值：

String - 32字节的16进制格式的交易哈希串。

如果交易是一个合约创建，请使用aiman.man.getTransactionReceipt()在交易完成后获取合约的地址。

RawTx =

{

"commitTime": 1548216476

"currency": "MAN"

"data": "0x"

"extra\_to": [{

"to": "MAN.3sJPZ6DzKntQCCQrR6q9U4d1bb48Q",

"value": "0x0de0b6b3a7640000"

}]

"gas": "0x033450"

"gasPrice": "0x0430e23400"

"isEntrustTx": 0

"lockHeight": 0

"nonce": "0x10000000000002"

"r": "0x79d67e927437a922ccc115f7611ebd047f582f21e701e88780297d47364a6754"

"s": "0x658331f3683945f2158d6c97364c50527c3cab1b1c886c564ff97751e05b48d8"

"to": "MAN.bjHU5ZCpKwg6k1nUCNG1vJKxVeBU"

"txEnterType": 0

"txType": 0

"v": "0x26"

"value": "0x0de0b6b3a7640000"}

Object - 要发送的交易对象。

* to: String - （可选）交易消息的目标地址，如果是合约创建，则不填.
* value: Number|String|BigNumber - （可选）交易携带的货币量，以zhu为单位。如果合约创建交易，则为初始的基金。
* gas: Number|String|BigNumber - 交易使用的gas，未使用的gas会退回。
* gasPrice: Number|String|BigNumber - 交易的gas价格 。
* data: String - （可选）或者包含相关数据的字节字符串，如果是合约创建，则是初始化要用到的代码。
* nonce: Number - 整数，使用此值，可以允许你覆盖你自己的相同nonce的，正在pending中的交易。
* V：签名结果
* R：签名结果
* S：签名结果
* txEnterType 交易入池类型（目前只有0）
* Currency：币种名称
* TxType：交易类型
* LockHeight：保留字段
* IsEntrustTx ：0-自付gas，1-代付gas
* CommitTime：提交时间，仅对定时和可撤销交易有效
* ExtraTo： 扩展交易（一对多交易填写）

返回值：

String - 32字节的16进制格式的交易哈希串

示例：

var rawtx= {

"to":"MAN.3amuUsMTakirpLt7uvS5pzsQp3ZPo",

"value":"0x69e679aec4a433900000",

"gasPrice":"0x430e23400",

"gas":"0x5208",

"data":"0x",

"nonce":"0x10000000000000",

"TxEnterType":0,

"IsEntrustTx":0,

"CommitTime":1547539401231,

"extra\_to":[],

"chainId":23,

"v":"0x52",

"r":"0x8fcb04c556e6412fd4273cea0ff26405dd56dfa604f354e8d7196d7e94cc4290",

"s":"0x1c6f2d8df3efb5228ebb180934df7a608f5d1a286c84fca4df1d2736ec06fff4",

"lockHeight":0,

"currency":"MAN",

"txType":0

}

aiman.man.sendRawTransaction(rawtx, **function**(err, hash) {

**if** (!err)

console.log(hash); *// "0x7f9fade1c0d57a7af66ab4ead79fade1c0d57a7af66ab4ead7c2c2eb7b11a91385"*

});

## aiman.man.call()

立刻执行一个新的消息调用，无需在区块链上创建交易

参数：

Object - 交易调用对象，信息结构如下：

from: STRING - 发送交易的原地址，可选

to: STRING - 交易目标地址

Currency:STRING-交易币种名称

gas: QUANTITY - 交易可用gas量，可选。man\_call不消耗gas，但是某些执行环节需要这个参数

gasPrice: QUANTITY - gas价格，可选

value: QUANTITY - 交易发送的MAN数量，可选

data: DATA - 方法签名和编码参数的哈希，可选

* QUANTITY|TAG - 整数块编号，或字符串"latest"、"earliest"

返回值

DATA - 所执行合约的返回值

示例：

var obj = {"to": "MAN.468kLTuAEjm53ro2pPErnAAHqbccK","data": "0x58975919","currency": "MAN"};

var result = aiman.man.call(obj,"latest");

console.log(result)

## aiman.man.sign

aiman.man.sign(address, dataToSign, [, callback])

使用指定帐户签名要发送的数据，帐户需要处于unlocked状态。

参数：

* String - 签名使用的地址
* String - 要签名的数据
* Function -（可选）回调函数，用于支持异步的方式执行[async]。

返回值：

String - 签名后的数据。

返回的值对应的是ECDSA（Elliptic Curve Digital Signature Algorithm）[12](http://web3.tryblockchain.org/Web3.js-api-refrence.html#fn12)签名后的字符串。

r = signature[0:64]

s = signature[64:128]

v = signature[128:130]

需要注意的是，如果你使用ecrecover，这里的v值是00或01，所以如果你想使用他们，你需要把这里的v值转成整数，再加上27。最终你要用的值将是27或28[13](http://web3.tryblockchain.org/Web3.js-api-refrence.html#fn13)。

示例：

var result = aiman.man.sign("0x135a7de83802408321b74c322f8558db1679ac20",

"0x9dd2c369a187b4e6b9c402f030e50743e619301ea62aa4c0737d4ef7e10a3d49"); // second argument **is** aiman.sha3("xyz")

console.log(result); // "0x30755ed65396facf86c53e6217c52b4daebe72aa4941d89635409de4c9c7f9466d4e9aaec7977f05e923889b33c0d0dd27d7226b6e6f56ce737465c5cfd04be400"

## aiman.man.estimateGas

aiman.man.estimateGas(callObject [, callback])

在节点的VM节点中执行一个消息调用，或交易。但是不会合入区块链中。返回使用的gas量。

参同aiman.man.sendTransaction，所有的属性都是可选的。

返回值：

Number - 模拟的call/transcation花费的gas。

示例：

**var**

var r = aiman.man.estimateGas({currency:"MAN"});

console.log(r);

aiman.man.estimateGas({ from: "MAN.3Hr2UCai2qz5FftusnJpn6dvrQGua",

to: "MAN.3Hr2UCai2qz5FftusnJpn6dvrQGua",

gas: "0x76c0a",

gasPrice: "0x9184e72a000",

value: "0xe",

nonce:"0x1000000000000f",

extraTo: [],

data: "0x",

currency:"MAN"

}

,function (err,res){

if (!err) {

console.log(res);

} else {

console.log(err);

}

});

## aiman.man.getDepositByAddr

返回指定账户的抵押信息

aiman.man.getDepositByAddr("MAN.4DnD4CZ1LNiHeb9SZNrn6XQfuETPC","latest")

### 参数

* String – 字符串，MAN地址
* QUANTITY|TAG - 整数块编号，或字符串"latest"、"earliest"

### 返回值

* OBJECT –抵押账户信息，信息结构如下：

AddressA0：STRING-抵押地址

AddressA1: STRING-签名地址

OnlineTime: QUANTITY-整形，在线时长，按照块高计数

Role：QUANTITY-参选身份, 128:验证者；16：矿工

PositionNonce: QUANTITY-整形，仓位nonce值

Dpstmsg：OBJECT-抵押仓位信息列表，抵押仓位信息结构如下：

DepositType：QUANTITY-整形，//0-活期,1-定期1个月,3-定期3个月,6-定期6个月

DepositAmount：QUANTITY-整形，以韦为单位的抵押值

Interest：QUANTITY-整形，以韦为单位的利息值

Slash：QUANTITY-整形，以韦为单位的惩罚值

BeginTime: QUANTITY-整形，UTC时间，抵押开始时间；活期为0

EndTime: QUANTITY-整形，UTC时间，定期表示退选时间；活期表示最早退款时间。

Position: QUANTITY-整形, 仓位号

WithDrawInfo：OBJECT-退款对象数组，退款对象信息结构如下：

WithDrawAmount：QUANTITY-整形，以韦为单位的退款金额

WithDrawTime：QUANTITY-整形，UTC时间，退款时间。

### 示例

请求：

var result = aiman.man.getDepositByAddr("MAN.4DnD4CZ1LNiHeb9SZNrn6XQfuETPC","latest")

result：

{

"AddressA0": "MAN.4Pn182LSJ3JNr9by4T5kDKsf127Jb",

"AddressA1": "MAN.Tfr4DZYNeWqqTth87phrg2KZcqya",

"OnlineTime": "0x81923",

"Role": "0x80",

"PositionNonce": 0,

"Dpstmsg": [

{

"DepositType": 0,

"DepositAmount": "0x8631d81fc94c92f9b336d",

"Interest": "0x2bf9f54df6717b96ef",

"Slash": "0x0",

"BeginTime": 0,

"EndTime": 0,

"Position": 0,

"WithDrawInfolist": []

}

]

}

## aiman.man.getDeposit

返回指定区块的抵押信息，通过高度指定区块

### 参数

QUANTITY|TAG - 整数块编号，或字符串"latest"、"earliest"

返回值：

* 抵押账户信息数组，信息结构如下：

Address：STRING-抵押地址

SignAddress: STRING-签名地址

Deposit: QUANTITY-抵押金额

WithdrawH: QUANTITY-参选状态，0, 在选；非0，退选高度

OnlineTime: QUANTITY-整形，在线时长，按照块高计数

Role：QUANTITY-参选身份, 128:验证者；16：矿工

示例：

var **var** result = aiman.man. getDeposit("latest"))；

[

{

"Address": "MAN.4M29vNSn7jG1dwLs7bxG2RHYLikWG",

"SignAddress": "MAN.3EX7HepVHi85wAh6duzGotYGbpBQR",

"Deposit": 1e+25,

"WithdrawH": 0,

"OnlineTime": 0,

"Role": 128

},

{

"Address": "MAN.439xNJTQrmmYEkm2wm5ThU79UsMiu",

"SignAddress": "MAN.4URp9YU7ufxNf1i1EbUs1PGhYwFzJ",

"Deposit": 1e+25,

"WithdrawH": 0,

"OnlineTime": 0,

"Role": 128

},

{

"Address": "MAN.2LrYy4EbhAi7Az4HdzKHbbyraMPBj",

"SignAddress": "MAN.4PMxVeKv9JZpBxK8oWcMDmcTUZtn9",

"Deposit": 1e+22,

"WithdrawH": 0,

"OnlineTime": 0,

"Role": 16

}

]

## aiman.man.getMatrixCoin

返回多币种信息。

aiman.man.getMatrixCoin('latest')

### 参数

* QUANTITY|TAG - 整数块编号，或者字符串"latest", "earliest"

### 返回值

* OBJECT – STRING数组，多币种名称

### 示例

请求：

var result = aiman.man.getMatrixCoin('latest');

响应：[

"MBN",

"MCN",

"MDN",

"MEN",

"MIN",

"MSN",

"MUSD"

]

## aiman.man.getMatrixCoinConfig

返回多币种的配置信息

aiman.man.getMatrixCoinConfig("MBN","latest")

### 参数

* STRING – 币种名称
* QUANTITY|TAG - 整数块编号，或者字符串"latest", "earliest"

### 返回值

OBJECT – 币种配置信息。结构如下：

* CoinRange：STRING。扩展保留。
* CoinType：STRING。支付币种名称。
* PackNum：QUANTITY。整形，每区块打包交易最大值。
* CoinUnit：QUANTITY。整形，币种单位。
* CoinTotal：QUANTITY。整形，币种发行量。
* CoinAddress：DATA，20字节。币种交易费账户地址。

### 示例

请求：

var result = aiman.man.getMatrixCoinConfig("MBN","latest")

返回：

[ {

"CoinRange": "MEN",

"CoinType": "MEN",

"PackNum": 9999,

"CoinUnit": "0xde0b6b3a7640000",

"CoinTotal": "0x33b2e3c9fd0803ce8000000",

"CoinAddress": "0x8000000000000000000000000000000000000001"

}

]

## aiman.man.getDestroyBalance

返回创建多币种的消耗MAN数量。

aiman.man.getDestroyBalance()

### 参数

无

### 返回值

QUANTITY - 整数，以zhu为单位的当前多币种发币价格

### 示例

请求：

var result = aiman.man.getDestroyBalance()

返回：

"0x152d02c7e14af6800000" // 100000000000000000000000

## aiman.man.getSignAccounts

返回指定区块的区块签名列表，通过Hash/块高

### 参数

块哈希或者块高

### 返回值

Object Array-签名对象数组。签名结构如下：

* Sign: DATA, 65字节 – 签名
* Account:STRING，–签名账户
* Validate: Boolean , -签名状态。TRUE：签名；FALSE：未签名
* Stock: QUANTITY ,股权。目前无效。

### 示例

请求：

var result = aiman.man.getSignAccounts(911);

var result1 = aiman.man.getSignAccounts("0x4bb2bb1ffc4e4f44b862c058fbafc948dd6942857368c76af34cb0fde777228e"));

返回：[

{

"sign": "0xf3fd53b57786bee28aa27e8439251e8c1c5233db964b8b69528ebc46861e79090f4ae49e617707c11e121e24ef099160262bcdf2347033d399d003ef7f6d69e100",

"account": "MAN.3pTAH8pGwmJ94Zm1B1sNMcm4q1kVr",

"validate": true,

"stock": 0

},

{

"sign": "0xcd577141ac1a9173316f20df9a39318a107169c5d93659a04b749cddb606e9f16e70b37cad5230c4e4a0e2c071f5d6b701dadf005211539b92309a9fc75e6c4501",

"account": "MAN.4Pn182LSJ3JNr9by4T5kDKsf127Jb",

"validate": true,

"stock": 0

},

{

"sign": "0xdc61de03cae70a4e010c695c123142e51aa9beff7b29be25c09cf784130558c634044c5d86f9c19898255d95494283a19ac77bb7ab59c96159537b6fa94a858f01",

"account": "MAN.23983gYFh8YH1ovpzKa3wpcNaRVn7",

"validate": true,

"stock": 0

}

]

## aiman.man.getValidatorGroupInfo

返回指定区块联合挖矿信息，通过块高指定区块。

aiman.man.getValidatorGroupInfo()

### 参数

* QUANTITY|TAG - 整数块编号，或字符串"earliest"、"latest"

### 返回值

OBJECT-联合挖矿信息MAP。MAP：以子合约地址为键值， STRING；合约存储信息为值的切片,OBJECT。

合约存储信息结构如下：

* OwnerInfo。OBJECT对象，合约创建信息，结构如下：

Owner:String- Owner账户地址

SignAddress：String-签名账户地址

WithdrawAllTime：QUANTITY-全部撤销时间

* Reward。OBJECT对象，回报分配信息，结构如下：

LevelRate：OBJECT对象数组-参与者门限和回报比例信息，结构如下：

Rate：对象。分配比例系数。{ Decimal ： 分母； Rate ：分子}；

Threshold：QUANTITY –整形，参与的最低门限。单位韦。

NodeRate：OBJECT对象-管理费信息，结构如下：

Decimal：QUANTITY –整形，管理费比例的分母

Rate：QUANTITY –整形，管理费比例的分子

OwnerRate：OBJECT对象数组-owner回报比例信息，结构如下：

Decimal：QUANTITY –整形，owner回报比例的分母

Rate：QUANTITY –整形，owner回报比例的分子

ValidatorMap : OBJECT, 参与者信息数组，包括Owner；参与者信息结构如下：

Address：STRING-参与者账户地址;

Reward: QUANTITY –整形,奖励分配累加值

AllAmount: QUANTITY –整形,总抵押值

Current: OBJECT-活期抵押信息，结构如下：

Amount: QUANTITY –整形,活期抵押金额

PreAmount: QUANTITY –整形,上一轮的活期抵押金额

Interest: QUANTITY –整形,活期利息

WithdrawList：退出抵押信息列表.a) WithDrawAmount 退出金额;b) WithDrawTime 退出时间

Positions : OBJECT-定期抵押仓信息列表;定期抵押信息结构如下：

DType: QUANTITY –整形,定期抵押类型

Position: QUANTITY –整形, 定期抵押仓号

Amount: QUANTITY –整形,定期抵押金额

EndTime: QUANTITY –整形,定期结束时间

### 示例

请求：

var result = aiman.man.getValidatorGroupInfo(“latest”);

返回：{

"MAN.w8Nvei3g98GAtbuCzPWHb9UiDRZj": {

"OwnerInfo": {

"Owner": "MAN.s7jf4pkbZdfnUaZyDL4EGZprWrHm",

"SignAddress": "MAN.4SYA8BxUZaevRXczJvdoSMT2p89Np",

"WithdrawAllTime": 0

},

"Reward": {

"LevelRate": [

{

"Rate": {

"Decimal": "1000000000",

"Rate": "1"

},

"Threshold": "0"

},

{

"Rate": {

"Decimal": "1000000000",

"Rate": "1"

},

"Threshold": "10000000000000000000000"

},

{

"Rate": {

"Decimal": "1000000000",

"Rate": "1"

},

"Threshold": "100000000000000000000000"

}

],

"NodeRate": {

"Decimal": "1000000000",

"Rate": "0"

},

"OwnerRate": {

"Decimal": "1000000000",

"Rate": "1"

}

},

"ValidatorMap": [

{

"Address": "MAN.B1dmKTmWqNfHpv6mKytRrCy2NkWi",

"AllAmount": "100000000000000000000",

"Current": {

"Amount": "100000000000000000000",

"Interest": "5891875148930089",

"PreAmount": "0",

"WithdrawList": []

},

"Positions": [],

"Reward": "57970186835851759"

},

{

"Address": "MAN.G8dYKuZvPQAqVXRSzCTV7sYPhGFK",

"AllAmount": "10000000000000000000000",

"Current": {

"Amount": "10000000000000000000000",

"Interest": "589187514893008924",

"PreAmount": "0",

"WithdrawList": []

},

"Positions": [],

"Reward": "0"

}

]

}

## aiman.man.getStorageAt

返回指定地址存储位置的值。

aiman.man.getStorageAt("MAN.468kLTuAEjm53ro2pPErnAAHqbccK",12,"MAN","latest")

### 参数

* DATA - 20字节，存储地址
* QUANTITY - 存储中的位置号
* CURRENCY – 币种
* QUANTITY|TAG - 整数块号，或字符串"latest"、"earliest"

### 返回值

DATA - 指定存储位置的值

### 示例代码

var result = aiman.man.getStorageAt("MAN.468kLTuAEjm53ro2pPErnAAHqbccK",12,"MAN","latest")

返回：

{"0x000000000000000000000000000000000000000000000000000000000000162e"}

## aiman.man.getTopologyStatusByNumber

获取指定区块顶层节点拓扑信息，通过高度指定区块。

aiman.man.getTopologyStatus(120)

### 参数

* QUANTITY|TAG - 整数块编号，或字符串"earliest"、"latest"

### 返回值

Object - 匹配的拓扑信息对象。结构如下：

* leader\_reelect：Boolean  -验证者重选状态。False:未发生验证者重选；True：发生验证者重选；
* validators：验证者信息数组。验证者信息结构如下：

account：STRING-账户；

online：Boolean  -在线状态。True:在线；False：离线

position：QUANTITY-位置编号

* backup\_validators：备份验证者信息数组。备份验证者信息结构见验证者信息结构
* miners：矿工信息数组。矿工信息结构见验证者信息结构
* elect\_validators：验证者主节点共识上下线信息。信息结构见验证者信息结构
* elect\_backup\_validators：验证者备份节点共识上下线信息。信息结构见验证者信息结构

### 示例

请求：

var result = aiman.man.getTopologyStatus(120);

响应：

{

"leader\_reelect": false,

"validators": [

{

"account": "MAN.4Pn182LSJ3JNr9by4T5kDKsf127Jb",

"online": true,

"position": 8192

},

{

"account": "MAN.23983gYFh8YH1ovpzKa3wpcNaRVn7",

"online": true,

"position": 8193

},

… //省略

{

"account": "MAN.2JJ1KuzNDAgfRRby6hfM7V256V1mD",

"online": true,

"position": 8210

}

],

"backup\_validators": [

{

"account": "MAN.31uRb2uAFrGvAoJwurJaZrNwpw6vn",

"online": true,

"position": 12288

},

…//省略

{

"account": "MAN.3ygAdEiWqv9U8pdQxw6T5C8DGqYqF",

"online": true,

"position": 12291

},

{

"account": "MAN.kaXBV56J1MayJzrcTQdm4LaGKA6a",

"online": true,

"position": 12292

}

],

"miners": [

{

"account": "MAN.253ZT6RZWK28o78gqAn2HkS73mcdN",

"online": true,

"position": 0

},

{

"account": "MAN.3pFY9DFmU3MWTFQeQRiCzs9bzKyqm",

"online": true,

"position": 1

},

…//省略

{

"account": "MAN.NcLaq3ZuHvuahrBn1tBNzGz55giv",

"online": true,

"position": 31

}

],

"elect\_validators": null,

"elect\_backup\_validators": [

{

"account": "MAN.2mHv8WkPczgZaoruUx9nn8b7dQaev",

"online": false,

"position": 61440

}

]

}

## aiman.man.getBlackList

返回交易黑名单账户

### 参数

无

### 返回值

OBJECT:字符串数组-加入黑名单的交易账户地址

### 示例

请求：

var result = aiman.man.getBlackList();

响应：

{"MAN.4Pn182LSJ3JNr9by4T5kDKsf127Jb"," MAN.Tfr4DZYNeWqqTth87phrg2KZcqya"}

备注：非共识数据，节点数据，链上无法验证。

## aiman.man.getSelfLevel

返回当前节点在网络中的身份.

### 参数

无

### 返回值

QUANTITY –身份标志。0：错误；1~4桶序号；5顶层节点；6普通节点

### 示例

请求：

console.log(aiman.man.getSelfLevel()); // 6

## aiman.man.getUpTime

返回指定账户指定区块的在线时长。通过块高指定区块。

### 参数

* STRING – 账户地址
* QUANTITY|TAG - 整数块编号，或字符串"earliest"、"latest"

### 返回值

QUANTITY – 整型值在线时长，按照块高计数。

### 示例

请求：

console.log(aiman.man.getUpTime("MAN.4Kq1ciivykmrCFzz5YjLgamcNdU76")) //0x85edb

## aiman.man.newBlockFilter

在节点中创建一个过滤器，以便当新块生成时进行通知。要检查状态是否变化， 请调用aiman.man.getFilterChanges。

### 参数

无

### 返回值

QUANTITY - 过滤器编号

### 示例代码

请求：

var result = aiman.man.newBlockFilter();

响应：

" 0x15842a94627e19e5571559236b5b1700 "

## aiman.man.newFilter

基于给定的选项创建一个过滤器对象，接收状态变化时的通知。要检查状态是否变化， 请调用aiman.man.getFilterChanges。

关于特定主题过滤器的说明：主题是顺序相关的。如果一个交易的日志有主题[A, B]，那么将被 以下的主题过滤器匹配：

* [] 任何主题
* [A] 先匹配A主题
* [null, B] 先匹配其他主题，再匹配B主题
* [A, B] 先匹配A主题，再匹配B主题，最后匹配其他主题
* [[A, B], [A, B]] "先匹配A主题或B主题，再匹配A主题或B主题，最后匹配其他主题

### 参数

Object - 过滤器选项对象：

* fromBlock: QUANTITY|TAG - 可选，默认值："latest"。16进制字符串块编号，或字符串"latesr"表示最后挖出的块，"earliest"用于未挖出的交易。
* toBlock: QUANTITY|TAG - 可选，默认值："latest"。16进制字符串块编号，或字符串"latesr"表示最后挖出的块，或"earliest"用于未挖出的交易。
* address: DATA|Array, 20字节 - 可选，合约地址或生成日志的一组地址
* topics: Array of DATA, - 可选，32字节主题数组，每个主题可以是数组或使用or选项连接

### 返回值

QUANTITY - 过滤器编号

### 示例代码

请求：

var para = JSON.parse('{"fromBlock":"0x0","toBlock":"latest","topics":[]}');

console.log(aiman.man.newFilter(para));

响应：

“0x15842a94627e19e5571559236b5b1700”

## aiman.man.uninstallFilter

写在具有指定编号的过滤器。当不在需要监听时，总是需要执行该调用。另外，过滤器 如果在一定时间内未接收到aiman.man.getFilterChanges调用会自动超时。

### 参数

QUANTITY - 过滤器编号

### 返回值

Boolean - 如果成功卸载则返回true，否则返回false

### 示例代码

请求：

var result aiman.man.uninstallFilter("0xb"); // true

## aiman.man.getFilterChanges

轮询指定的过滤器，并返回自上次轮询之后新生成的日志数组。

### 参数

QUANTITY - 过滤器编号

### 返回值

Array - 日志对象数组，如果没有新生成的日志，则返回空数组

使用aiman.man.newBlockFilter创建的过滤器将返回块哈希（32字节），例如["0x3454645634534..."]。

使用aiman.man.newFilter创建的过滤器，日志对象具有如下参数：

* removed: TAG - 如果日志已被删除则返回true，如果是有效日志则返回false
* logIndex: QUANTITY - 日志在块内的索引序号。对于挂起日志，该值为null
* transactionIndex: QUANTITY - 创建日志的交易索引序号，对于挂起日志，该值为null
* transactionHash: DATA, 32字节 - 创建该日志的交易的哈希。对于挂起日志，该值为null
* blockHash: DATA, 32字节 - 该日志所在块的哈希。对于挂起日志，该值为null
* blockNumber: QUANTITY - 该日志所在块的编号。对于挂起日志，该值为null
* address: DATA, 20字节 - 该日志的源地址
* data: DATA - 包含该日志的一个或多个32字节无索引参数
* topics: Array of DATA -0~4个32字节索引日志参数的数据。在solidity中，第一个主题是事件签名，例如 Deposit(address,bytes32,uint256)，除非你声明的是匿名事件

### 示例代码

请求：

var result = aiman.man.getFilterChanges("0x16");

响应：

[{

"logIndex": "0x1", // 1

"blockNumber":"0x1b4", // 436

"blockHash": "0x8216c5785ac562ff41e2dcfdf5785ac562ff41e2dcfdf829c5a142f1fccd7d",

"transactionHash": "0xdf829c5a142f1fccd7d8216c5785ac562ff41e2dcfdf5785ac562ff41e2dcf",

"transactionIndex": "0x0", // 0

"address": "0x16c5785ac562ff41e2dcfdf829c5a142f1fccd7d",

"data":"0x0000000000000000000000000000000000000000000000000000000000000000",

"topics": ["0x59ebeb90bc63057b6515673c3ecf9438e5058bca0f92585014eced636878c9a5"]

},{

...

}]

## aiman.man.getFilterLogs

返回指定编号过滤器中的全部日志。

### 参数

QUANTITY - 过滤器编号

### 返回值

请参阅aiman.man.getFilterChanges。

### 示例代码

请求：

var result = aiman.man.getFilterLogs"0x16");

响应请参阅aiman.man.getFilterChanges。

## Aiman.man.getLogs

返回指定过滤器中的所有日志。

### 参数

Object - 过滤器对象，参考aiman.man.newFilter调用的参数

### 返回值

请参考aiman.man.getFilterChanges调用。

### 示例代码

请求：

var result = aiman.man.getLogs("0x000000000000000000000000a94f5374fce5edbc8e2a8697c15331677e6ebf0b");

响应请参考aiman.man.getFilterChanges。

## Aiman.man.getEntrustList

根据指定授权账户所有最新区块的委托信息。

### 参数

STRING- 授权账户地址。

### 返回值

OBJECT-委托账户信息列表。委托账户信息结构如下：

* EntrustAddres : STRING-委托账户地址
* IsEntrustGas：Boolean  -是否委托GAS代付。True表示是委托Gas代付。
* IsEntrustSign：Boolean  -是否委托签名。True表示是委托签名。
* EnstrustSetType ：DATA， 1byte- 0：按高度委托,1：按时间委托,2：按次数委托
* StartHeight ：QUANTITY - 整数块编号，委托的起始高度
* EndHeight：QUANTITY - 整数块编号，委托的结束高度
* StartTime：QUANTITY – UTC时间，委托的起始时间
* EndTime：QUANTITY – UTC时间，委托的结束时间
* EntrustCount：QUANTITY-委托次数

### 示例代码

请求：

var result = aiman.man.getEntrustList("MAN.21LDRQtTVBdv3EcktxrVQebpQmLEA”);

返回[

{

"EntrustAddres": "MAN.CUVXBNBpqXipn5JtDsnWKz3e59to",

"IsEntrustGas":false,

"IsEntrustSign":false,

"EnstrustSetType":2,

"StartHeight":0,

"EndHeight":0,

"StartTime":0,

"EndTime":0,

"EntrustCount": 5

}

]