# I/O COIN

# IOC API Stealth (V.2)

# API technical documentation (version 1.6 - Codename : GREEN) – April 17th 2018

To access the I/O Coin Blockchain you need to download the latest files from Github and synchronize the full Blockchain. Blockchain files for faster sync are provided in Github. To use the available features you should have some IOC in your wallet to pay for the services.

The IOC you pay for the data services will be redistributed to the active stakers in the network. If you need assistance or want to share your ideas, thoughts and solutions, please connect to our Telegram Developer Channel

- Get Info
- Get Help
- Get Block Count
- RSA Keys
- Check Wallet
- Repair Wallet
- Alias Registration
- Alias Decryption
- Upload Data
- Download Data
- Transfer Alias
- Transfer Encrypted Alias
- Send Public Key
- Status List
- Send Message
- Received Messages

# GET INFO

```
iocoind getinfo
```

Prints various information about the node and the network.

## EXAMPLE

```
getinfo
{
  "version" : "v5.0.0.0-g6b2edb9",
  "protocolversion" : 60021,
  "walletversion" : 60000,
  "balance" : 0.51830000,
  ...
}
```

# GET HELP

```
iocoind help
```

Lists all available public RPC commands.

## EXAMPLE

```
help
[
  "addmultisigaddress",
  "addredeemscript",
  "addresstodion",
  "alias",
  ...
]
```

# GET BLOCK COUNT

```
iocoind getblockcount
```

Get the current, latest synced, block.

### EXAMPLE

```
getblockcount
1741667
```

# RSA KEYS

```
iocoind myrsakeys
```

Get a list of available wallet addresses and aliasses attached to it.

### EXAMPLE

```
myrsakeys
[
    {
        "address" : "iVEcVcbaNGf755dYPsR1eKnAzZhdcmbepK",
        "alias" : "NONE"
    },
    {
        "address" : "ipFAdkYm9JD3UsoFCfUnCqHvKdGkDkDHsr",
        "alias" : "dennuz666"
    },
    ...
```

# CHECK WALLET

```
iocoind checkwallet
```

Get the ‘health status’ of the wallet. If you are missing coins or a RPC gone wrong you can check the status here.

### EXAMPLE

```
checkwallet
{
    "mismatched spent coins" : 2,
    "amount in question" : 0.49830000
}
```

# REPAIR WALLET

```
iocoind repairwallet
```

If wallet check fails (gives an error status) you can try repair the wallet with this RPC.

### EXAMPLE

```
repairwallet
{
    "mismatched spent coins" : 2,
    "amount affected by repair" : 0.49830000
}
```

# ALIAS REGISTRATION

```
iocoind alias "alias"
```

Registers a new encrypted alias and returns the transaction id. The alias remains valid at the time of writing for 210000 blocks from any given update. Data fees are currently 0.01 IOC minimum plus 0.01 IOC for every 1Kb.

| PARAMETERS | |
|---|---|
| Required parameters | |
| alias | string |

### EXAMPLE

```
alias "my alias"
[
    "aa5d7b8d74d9e69ef4d029dd12fe635bed91f388e3c026d10b8fc1715f925108"
]
```

# ALIAS DECRYPTION

```
iocoind decryptAlias "alias" "associated_address"
```

Decrypt an alias with associated key address. You can get the associated address by using the statusList RPC. Data fees for decypting an alias are currently 0.01 IOC.

| PARAMETERS | |
|---|---|
| Required parameters | |
| alias | string |

| | |
|---|---|
| associated_address | string |

### EXAMPLE

```
decryptAlias "myaliastodecrypt" ipmPfW87XWUGEGufivjfHxd1VWed35hfCG
7c63ebf7ed21a8583819b2f581f72d2bd7949f3d379537193d37d7013478a142
```

# UPLOAD DATA

```
iocoind updateEncrypt "alias" "file_path"
```

The updateEncypt command associates data with the given (encrytped) alias. The data is extracted, compressed and encrypted on the blockchain using AES 256 encryption. You can check the encrypted status (boolean) of an alias in the wallet or use the ‘statusList’ command in the daemon. Data fees are currently 0.02 IOC.

| PARAMETERS | |
|---|---|
| Required parameters | |
| alias | string |
| file_path | stringLocal device path to the file. |

### EXAMPLE

```
updateEncrypt myalias C:\Users\username\Desktop\test.txt
fade2f2c3bb2e2d0877266bd119e89fff73f841a96dff4c405cc7fbc25f477d6
```

# DOWNLOAD DATA

```
iocoind downloadDecrypt "alias" "file_path"
```

Downloading data is possible in two ways. You can download data by alias, just like you uploaded it. After an alias is expired (see ‘Alias Registration’ for more info about expiration) you can then download data with the private key from the expired alias with the ‘downloadDecryptEPID’ command. You can get the address of an alias by using the statusList RPC.

| **PARAMETERS** | |
|---|---|
| Required parameters | |
| alias | string |
| file_path | stringLocal device storage path to save the file. |

### EXAMPLE

```
downloadDecrypt myalias C:\Users\username\Desktop\test.txt
   true
```

# DOWNLOADDECRYPTEPID

```
iocoind downloadDecryptEPID "address" "file_path"
```

| PARAMETERS | |
|---|---|
| Required parameters | |
| address | string |
| file_path | stringLocal device storage path to save the file. |

**EXAMPLE**

```
downloadDecryptEPID ib388VZfBNnbrJv86PVXeeUwAF8BZdKKGs C:\Users\username\Desktop\test.txt
true
```

# TRANSFER ALIAS

```
iocoind transferAlias "alias" "target"
```

Transfer a given alias to the target address or alias. Data fees are currently 0.01 IOC.

| PARAMETERS | |
|---|---|
| Required parameters | |
| alias | string |
| target | string Target alias or address. |

### EXAMPLES

```
transferAlias myaliastodecrypt ikQHhA73EnmBsoV7ZBxrZYzPK5WtAM9UXR
```

```
transferAlias myaliastotransfer targetalias
```

# TRANSFER ENCRYPTED ALIAS

```
iocoind transferEncryptedAlias "alias to transfer" "alias address to transfer" "target wallet address"
```

Transfer an encrypted alias over a channel. You can get the alias address by using the statuslist command. Data fees are currently 0.02 IOC.

| PARAMETERS |
|---|
| Required parameters |

| | |
|---|---|
| alias | string |
| address | stringThe address of the alias. |
| address | stringTarget wallet address. |

### EXAMPLE

```
transferEncryptedAlias myalias iWTgoUFgAyBCbAfbLB9qXQBn1RsHhHrWir
ikQHhA73EnmBsoV7ZBxrZYzPK5WtAM9UXR
    fade2f2c3bb2e2d0877266bd119e89fff73f841a96dff4c405cc7fbc25f477d6
```

# SEND PUBLIC KEY

```
iocoind sendPublicKey "address" "address"
```

To send or receive an alias users would have to send an invite From Public alias (A) to Public alias (B) as in an rsa key exchange. This would initiate an encrypted tunnel, giving the ability to transfer aliases, but also initiate messaging between users.

| PARAMETERS |
|---|

| Required parameters | |
|---|---|
| address | stringThe address of the local alias. |
| address | stringThe address of the target alias. |

**EXAMPLE**

```
sendPublicKey ipFAdkYm9JD3UsoFCfUnCqHvKdGkDkDHsr ifURRvG6asxQJauo9G8gGEfqnbYKzq3UYK
[
    "253fce5a099fe97322fcb39d6f5028a509d76405fea6852ac2d37cd5926ebc9f",<

"IH4+eeUgKHqEIWBfk9OH5rEwHA2T8a+iuPqmDgKmmGgBO4QmEHXwBf7Rub4bATf1yXjX31rbQXe2dm2A47oDK
B8="
]
```

# STATUS LIST

```
iocoind statusList ["alias"]
```

List all I/O Coin addresses that have RSA key pairs associated with them. You can check the status of an alias with this command by adding the optional alias parameter. The wallet needs to be unlocked.

| **PARAMETERS** |
|---|

| Optional parameters | |
|---|---|
| alias | stringoptional paramter only for unencrypted aliasses. |

### EXAMPLE

```
statusList myalias
[{
    "alias" : "myalias",
    "encrypted" : "true",
    "address" : "iWmHymK9bDSQeAJmoFzafwiMRPbjyfHDfC",
    "nHeigt" : 1733471,
    "expires_in" : 209729,
    "xtuVector" : "transform"
}]
```

# SEND MESSAGE

```
iocoind sendMessage "from address" "message" "to address"
```

Send an encrypted message over an established channel. Read Send Public Key for setting up an established channel.

| **PARAMETERS** |
|---|
| Required parameters |

| | |
|---|---|
| from address | stringThe address of the local alias. |
| message | stringThe message to send. |
| to address | stringThe address of the target alias. |

### EXAMPLE

```
sendMessage ipFAdkYm9JD3UsoFCfUnCqHvKdGkDkDHsr "Test" ifURRvG6asxQJauo9G8gGEfqnbYKzq3UYK
    fade2f2c3bb2e2d0877266bd119e89fff73f841a96dff4c405cc7fbc25f477d6
}]
```

# RECEIVED MESSAGES

```
iocoind decryptedMessageList
```

List the received encrypted messages. Encrypted messages are locally decrypted for viewing in plain text. Needs an unlocked wallet.

### EXAMPLE

```
decryptedMessageList
[{
    "sender" : "ipFAdkYm9JD3UsoFCfUnCqHvKdGkDkDHsr",
    "recipient" : "ifURRvG6asxQJauo9G8gGEfqnbYKzq3UYK",
    "encrypted_message" : "Klt7ch/FyubdWlGG6Jj1Xg==",
    "time" : "2018-04-12 05:20:48 UTC",
    "plain_text" : "Test",
    "iv128Base64" : "vlYaWjucdK3aOtZGjZsK2g==",
    "signature" :
"H0ZJ43IefH6LeKzrLMiBMrfdQvnXSC013czsbAWF3NrbJIwVNL4+qdBLf5KCGVPBIVcjjqlVsNSFfipy0jD85+0="
}]
```

# Stealth group

The following calls allow for stealth sending of funds to a so called stealth address hereafter known as ***shade*** address or simply a ***shade*** . Shades encapsulate what are known hereafter as abs and ord keys.

```
locoind shade
```

Generates a shade address.

## EXAMPLE

```
liocoind shade
```

```
[    "mi6Sv2ow1ywQgRF7P2ueH6U5Ykii1rCG8c",

   "mtqcQSz5vpqEa4fuQaH1gJxGy6K6RdRpYX",
"iH9fMZjDrVp8MY7S5cZcTTV5btT8zPnGYqnQx7BRjvm9aySYXdJTfYtECMrr5eNgFvDT1JvTTY1SX1uigbBv2cjenG
m"]
```

| OUTPUT  PARAMETERS | |
|---|---|
| The abs key | mi6Sv2ow1ywQgRF7P2ueH6U5Ykii1rCG8c |

| | |
|---|---|
| The ord key | mtqcQSz5vpqEa4fuQaH1gJxGy6K6RdRpYX |
| The resulting shade | iH9fMZjDrVp8MY7S5cZcTTV5btT8zPnGYqnQx7BRjvm9aySYXdJTfYtECMrr5eNgFvDT1JvTTY1SX1uigbBv2cjenGm |

The shade address is subsequently used by the receiver.

```
locoind shadesend <shade> <amount>
```

| **MANDATORY INPUT PARAMETERS** | |
|---|---|
| shade | (string) The target shade address for the stealth send |
| amount | (double) The amount to send |

| **OUTPUT INFORMATIONAL PARAMETERS** | |
|---|---|
| abs | (string) The recovered abs key |
| ord | (string) The recovered ord key |
| target | (string) The one-time end target receiving address |
| trace | (string) The reference trace key |
| txid | (string) The transaction id |

## EXAMPLE

Taking the previously generated shade from above, an associate may now perform stealth send :

```
iocoind  shadesend
iH9fMZjDrVp8MY7S5cZcTTV5btT8zPnGYqnQx7BRjvm9aySYXdJTfYtECMrr5eNgFvDT1JvTTY1SX1uigbBv2cjenGm
10

[

    {

        "abs" : "mi6Sv2ow1ywQgRF7P2ueH6U5Ykii1rCG8c",

        "ord" : "mtqcQSz5vpqEa4fuQaH1gJxGy6K6RdRpYX",

        "target" : "mm9PnZRGWVRWAdiTxZTtWRbx42MCyUsgxW",

        "trace" : "n1Xi7itgxnVgbAw7jA6WepMP26dMr5BJjQ",

        "txid" : "b59acd8eb59cf753414e5e064a39041815fb3779a88f34120d06da148401d717"

    }

]
```

## Known projects

Any project can start developing and using the IOC Blockchain for their project. The below projects are in development by external teams.

Want your project listed? Contact us.