

Timekoin API Specification

This document contains all the technical information to allow someone to write external applications that can interface and poll functions and events within the Timekoin server. All request are a combination of GET and POST request with the output specified for each API poll. Provided the proper access is granted to that function for the HASH code used during the polling.

Example:

To ask the server what the balance is of a public key, the request would be formed like this provided you know the server Domain/IP, the port number (if other than 80), and sub-folder location (timekoin for example):

http://myserver:80/timekoin/api.php?action=pk_balance&hash=MyHashCodeHere

POST DATA NAME = [public_key]

LS0tLS1CRUdJTiBQVUJMSUMgS0VZLS0tLS0KTUIIzk1BMEdDU3FHU0liM0RRRUJBUVVBQTRITkFEQ0J5UUtCd1FEQ0hlbmZ1bGllTDZpRENUUzJLYTBrUFdPY095TEZhT2JWUFRWTQpCN3IGdTUzRnJPRWZlY2tDRUtwdDRhWGZGRjl0emVKd0ptZXAwNEcxcHRKcisxZ2JHNUhoako0YUxKc3MxL3crVmJnemhPZXBpLzRfCkxBR0c5OEadmS2FKQkRMSkhrRHFtdCsrU1hTYzBmdVdCWk8rTjllU3dNVVVCNmpIN21FMS81emhxNkY2M1gzVHFYSGNCdW02ZUtYYzQKNG5qY2syWXYVZk9zaE80WkN5VnFGOTJEUeEhOQTVncit1SmdSYVpoeWdLTXVpMDd2SnRZdGpyQ2I2YVlPTEhwNENwVUNBd0VBQVE9PQotLS0tLUVORCBQVUJMSUMgS0VZLS0tLS0=

Result: 5

Public Key Actions

Action

pk_balance

Parameters

Timekoin Public Key in POST Data field named **public_key** in Base64 Encoded Format

Description

Returns the balance for the provided public key as whole numbers with no formatting. Example: 1, 55, or 9999
If the key does not exist, then 0 will be returned.

If the key does exist and happens to have a balance of 0, then 0 will also be returned.

Action

pk_gen_amt

Parameters

Timekoin Public Key in POST Data field named **public_key** in Base64 Encoded Format

Description

Returns the amount of Timekoins currently being generated by the provided public key or 0 if the public key is not in the generation list.

Action

pk_history

Parameters

- Timekoin Public Key in POST Data field named **public_key** in Base64 Encoded Format
- How many transactions from the transactions history to retrieve in POST Data field named last in whole Integer Format with a Maximum value of 100 and minimum value of 1. Any values higher or lower than the range will be considered just 1
- Retrieve transactions sent to the public key is specified by using the POST Data field with the name **sent_to** set to a value of 1 (or TRUE)
- Retrieve transactions sent from the public key is specified by using the POST Data field with the name **sent_from** set to a value of 1 (or TRUE)

Description

Returns the transaction history of a public key in the following format.

If neither field of **sent_to** or **sent_from** is set, the action will fail and return 0 (or FALSE)

Only one transaction direction per query can be done. You can not mix **sent_to** and **sent_from** at the same time.

If the public key is not found, the action will fail and no data will be returned.

Example Output Format

```
---TIMESTAMP=<timestamp of transaction>
---FROM=<public key that sent the transaction to the public key being queried>
---TO=<public key that was sent a transaction from the public key being queried>
---AMOUNT=<amount of transaction>
---VERIFY=<verification level>
---MESSAGE=<any encoded message in transaction>---END
```

The newest transaction starts with 1 and others follow in order as 2,3,4 until the amounts being queries are complete.

Example output

---TIMESTAMP1=1234567890	---TIMESTAMP2=1234567892
---FROM1=<public key in base64>	---FROM2=<public key in base64>
---TO1=<public key in base64>	---TO2=<public key in base64>
---AMOUNT1=55	---AMOUNT2=3
---VERIFY1=22	---VERIFY2=127
---MESSAGE1=thanks for the help---END1	---MESSAGE2=---END2

Action

pk_valid

Parameters

Timekoin Public Key in POST Data field named **public_key** in Base64 Encoded Format

Description

Returns 1 (or TRUE) if the Public Key has any kind of history of transactions associated with. Returns 0 (or FALSE) if the Public Key has no history of transactions associated with it. This is a way to test for a valid Public Key before a transaction is sent, but can not tell the user if a Public Key that has been created, but never used, is valid or not.

Action

pk_recv

Parameters

Timekoin Public Key in POST Data field named **public_key** in Base64 Encoded Format

Description

Returns a total of *all* the Timekoins ever received by the provided public key through transactions.

Action

pk_sent

Parameters

Timekoin Public Key in POST Data field named **public_key** in Base64 Encoded Format

Description

Returns a total of *all* the Timekoins ever sent out by the provided public key through transactions.

Action

pk_gen_total

Parameters

Timekoin Public Key in POST Data field named **public_key** in Base64 Encoded Format

Description

Returns a total of *all* the Timekoins ever generated by the public key.

Transaction Functions

Action

send_tk

Parameters

- Timestamp in POST Data field named **timestamp** in whole Integer Unix Time Format
- Public Key in POST Data field named **public_key** in Base64 Encoded Format
- Crypt1 Data in POST Data field named **crypt_data1** in Base64 Encoded Format
- Crypt2 Data in POST Data field named **crypt_data2** in Base64 Encoded Format
- Crypt3 Data in POST Data field named **crypt_data3** in Base64 Encoded Format
- Transaction SHA Hash Data in POST Data field named **hash**
- Transaction Attribute Data in POST Data field named **attribute**
- Qhash Data in POST Data field name **qhash**

Description

Saves a copy of a transaction directly into the “my_transaction_queue” table of the server. The server will then attempt to submit the transaction to the Timekoin network as though it had created the transaction itself. This has the benefit of the server keeping track of the transaction until it is saved in the transaction history.

Status Actions

Action

tk_hash_status

Parameters

Hashcode in GET request

Description

Returns TRUE (1) if the Hashcode is valid for any type of usage and NULL if the Hashcode is invalid.

Action

tk_trans_total

Parameters

- Total Transaction Cycles in GET request – **&last**=<whole Integer number>
- **&last**=<number> Maximum value of 100 and minimum value of 1. Any values higher or lower than the range will be considered just 1

Description

Returns how many transactions (standard and generation transactions) in string format.

Example Output Format

---TIMESTAMP=<timestamp of transaction cycle>

---NUM=<number of transactions>

---AMOUNT=<total amount of transactions sum together>---END

The newest transaction starts with 1 and others follow in order as 2,3,4 until the amounts being queries are complete.

Example output

---TIMESTAMP1=1234567890	---TIMESTAMP2=1234567891
---NUM1=22	---NUM2=22
---AMOUNT1=155---END1	---AMOUNT2=155---END2

Action

tk_process_status

Parameters

Which process to check in GET request

- **&process=1** – Main Program Process
- **&process=2** – Treasurer Processor
- **&process=3** – Peer Processor
- **&process=4** – Transaction Queue Clerk
- **&process=5** – Generation Peer Manager
- **&process=6** – Generation Processor
- **&process=7** – Transaction Clerk
- **&process=8** – Foundation Manager
- **&process=9** – Balance Indexer
- **&process=10** – Watchdog

Description

This will compare processing times between the current server time and the last update from the process. If the difference in time is greater than the allowed timeout, the process is considered either stalled or stopped completely. Returns TRUE (1) for running process and FALSE (0) for offline process.

Action

tk_start_stop

Parameters

Which system process to send commands in GET request

- **&active=1** – Start Main Program Process
- **&active=2** – Stop Main Program Process
- **&active=3** – Start Watchdog Process
- **&active=4** – Stop Watchdog Process

Description

This will send commands to the Timekoin server that will either start or stop the selected process of the system.

Returns TRUE (1) for successful start command.

Returns FALSE (0) for failure.

Returns (2) If the command sent is already complete. Example, sending Start command after the process has already started.

Easy Key Actions

Action

easy_key

Parameters

Easy Key Shortcut in POST Data field named **easy_key** in Base64 Encoded Format

Description

Returns the **public key** associated with the easy key from the transaction history in Base64 Encoded Format. If no easy key is found, then it will return a 0.

Action

num_gen_peers

Parameters

Use a 0 or 1 in GET request

- **&distinct=0** – Returns Number of Total Generating Public Keys
- **&distinct=1** – Returns Number of Unique Generating Public Keys
- **&public_keys=1** – Returns All Generating Public Keys List

Description

Returns the total number of public keys generating currency within the Timekoin network. Using the **distinct=1** GET request will narrow the number down to unique public keys only. Public keys that are on both the IPv4 and IPv6 Internet will count as 2 generating servers normally.

Using the **public_keys=1** returns a list output of all the public keys generating currency in Base64 Encoded Format using your distinction settings as the filter.

Example Output

---GEN_PUBLIC1=<Base64 Encoded Public Key> ---END1	---GEN_PUBLIC2=<Base64 Encoded Public Key> ---END2
---	---