Preliminary Factom API version 1.0

PDF created 24 Aug 2015

Method Name	Purpose
commit-chain	pay to create a new Chain with a specific first Entry
reveal-chain	publish an Entry to be the first Entry in a new Chain
commit-entry	pay to create a new Entry in existing Chain
reveal-entry	publish a committed Entry
directory-block-head	retrieve the KeyMR of the latest Directory Block
directory-block-by-keymr	retrieve a Directory Block given a KeyMR
entry-block-by-keymr	retrieve an Entry Block given a KeyMR
entry-by-hash	retrieve an Entry's contents given an Entry Hash
chain-head	retrieve the KeyMR of the latest Entry Block with the specified ChainID
entry-credit-balance	retrieve the number of Entry Credits allocated to the specified pubkey

commit-chain

POST /v1/commit-chain

json object to post:
{
	“CommitChainMsg”:”hex encoded Commit Chain message”
}

return http 200 on success

reveal-chain

POST /v1/reveal-chain

json object to post:
{
	“Entry”:”hex encoded binary Entry”
}

return http 200 on success

commit-entry

POST /v1/commit-entry

post json object:
{
	“CommitEntryMsg”:”hex encoded Entry commit message”
}

return http 200 on success

reveal-entry

POST /v1/reveal-entry

json object to post:
{
	“Entry”:”hex encoded binary Entry”
}

return http 200 on success

directory-block-head

GET /v1/directory-block-head

```
{
        “KeyMR”:”hex encoded Merkle root”
}
```

Returns a JSON object containing the KeyMR of the latest Directory Block. The KeyMR can be used to call directory-block-by-keymr which will return the latest block.

directory-block-by-keymr

GET /v1/directory-block-by-keymr/"hex encoded KeyMR"

```
{
        "Header":{
                "PrevBlockKeyMR":"hex encoded KeyMR of the previous Directory Block",
                "Timestamp":unix time (number),
                "SequenceNumber":number of previous Directory Blocks
        },
        "EntryBlockList":[                                              # three or more
                {
                        "ChainID":"hex encoded string"
                        "KeyMR":"hex encoded KeyMR",
                },
                {
                        "KeyMR":"hex encoded merkle root"
                        "ChainID":"hex encoded string",
                }
        ],
}
```

- Header
  - PrevBlockKeyMR - This is a 32 byte number which can be used to call directory-block-by-keymr again to get the previous block. The genesis block has a PrevBlockKeyMR of all zeros.
  - Timestamp - This is UTC time represented by unix time with 1 second LSB resolution. The time is when the Block was started. The blocks start every 10 minutes on the 10 minute mark.
  - SequenceNumber - The number of previous Directory Blocks is shown by this number.
- EntryBlockList - This is a list of all the Chains which had Entries added during the 10 minute window covered by the block.
  - ChainID - An identifier for each Chain, which users place Entries into. All Entries must specify one and only one Chain to go into. If a specific ChainID has one or more Entries added, then its ChainID will be listed.
  - KeyMR - This value can be used when calling entry-block-by-keymr to get all the Entries added during the 10 minute period.

entry-block-by-keymr

GET /v1/entry-block-by-keymr/”hex encoded KeyMR”

will return a single json object

```
{
        "Header":{
                "BlockSequenceNumber":number,
                "PrevKeyMR":"hex encoded KeyMR",
                "Timestamp":number(unix time)
                “ChainID”:”hex encoded ChainID”
        },
        "EntryList":[                                   # one or more
                {
                        "Timestamp":number,
                        "EntryHash":"hex encoded hash"
                },
                {
                        "Timestamp":number,
                        "EntryHash":"hex encoded hash"
                }
        ],
}
```

- Header
  - BlockSequence - This is the count of the Entry Blocks with this ChainID which have preceded this block.
  - PrevKeyMR - This is an identifier which can be used to request the previous Entry Block of this ChainID. If it is the first Entry Block, then this field is all zeros.
  - Timestamp - This is the timestamp of Directory Block which this Entry Block is contained in. It is UTC time represented by unix time with 1 second LSB resolution. The time is when the Block was started. The blocks start every 10 minutes on the 10 minute mark.
  - ChainID - This is a 32 byte unique value which represents this Chain. All the Entries in this Entry Block will share this ChainID. The ChainID is a series of SHA256 hashes of the Chain Name. See here.
- EntryList
  - Timestamp - This is a unix time which the Entry was acknowledged before. Factom has 1 minute resolution of timestamping. The data returned is with 1 second LSB resolution, but snaps to the next whole minute. Due to network propagation time, etc, the Entry may have been created before this time.

- EntryHash - This is a 32 byte identifier which is unique to the data in this Entry. It can be used to retrieve the Entry data with entry-by-hash. Duplicate Entries are allowed, and will share the same EntryHash. The same payload or content with a different ChainID will have a different EntryHash.

entry-by-hash

GET /v1/entry-by-hash/”hex encoded Entry Hash”

will return a single json object

```
{
        "ChainID":"hex encoded ChainID",
        "ExtIDs":[                              # zero or more
                "hex encoded string",
                "hex encoded string",
                ...
        ],
        "Content":"hex encoded data"
}
```

- ChainID - This is a 32 byte unique value which represents the Chain the Entry belongs to. The ChainID is a series of SHA256 hashes of the Chain Name. See here.
- ExtIDs - These are extra pieces of data which have defined lengths. They are intended to be used as external database keys to search against. Their validity is only enforced in the first Entry. The first Entry’s ExtIDs are interpreted as the Chain Name. They must hash to the ChainID.
- Content - This is the part of the Entry which the bulk of user data is held. It is open for interpretation by the user applications.

chain-head

GET /v1/chain-head/”hex encoded ChainID”

```
{
        “ChainHead”:”hex encoded KeyMR (or hash)”
}
```

This returns a way to get the latest block of the specified ChainID. Typically give it a ChainID and it returns KeyMR of the latest Entry Block. The returned value can be passed to entry-block-by-keymr to get the latest Entry Block contents. Advanced users can ask for these ChainIDs:

000000000000000000000000000000000000000000000000000000000000000f Factoid Chain head
000000000000000000000000000000000000000000000000000000000000000c Entry Credit chain head
000000000000000000000000000000000000000000000000000000000000000a Administrative Chain head

These can be used with the binary-blob call to get the serialized Factoid blocks, etc.

entry-credit-balance

GET /v1/entry-credit-balance/”hex encoded Entry Credit public key”

```
{
        “Balance”:number of Entry Credits
}
```

This is the current number of Entry Credits as interpreted by the local node. It will reflect the balance immediately deducted by usage, but balance increases must be acknowledged.