JSON Specification for Universal Wallet

Version 1.0.1

8/01/2022

Introduction

The purpose of this document is to define a JSON schema for the purpose of adding new currencies to the Open-Transactions wallet. The ultimate goal is the ability to add a new blockchain, token, or off-chain currency to the wallet without necessitating a recompile of the software.

For off-chain currencies and on-blockchain tokens, this is relatively easy, and the JSON schemas are provided within this document.

Difficulties

However, adding new blockchains is inordinately difficult, and some would even say impossible. Some examples of these difficulties may be illustrative:

- BCH uses a different signing algorithm than BTC. If the wallet source code does not already support this new signing algorithm, then it could not be added through JSON except as scripted code in the JSON data, which does not get around the requirement of adding new code, and which presents security risks of its own.
- PKT, though in most respects a clone of BTC, uses an entirely different block format and thus requires entirely different code to parse its blocks. Even if this code were to be added as a piece of script in the JSON — which presents the same risks described above for BCH — it would not solve the problem that the parsed data would then need to be placed into a new data structure located in the compiled code.
- Some blockchains introduce new proof-of-work algorithms. Dogecoin is one example of this. Monero is a more extreme example since it changes this algorithm on a regular basis (by design).
- Blockchains may also switch from standalone proof-of-work to merged-

mining.

- The fundamental differences between UTXO and Account-based blockchains.
- Sometimes blockchains change their rules at a certain block height.
- Sometimes, furthermore, the old rules continue in operation alongside the new rules, producing a fork where one blockchain has now become two blockchains operating simultaneously.
- When this happens, all the tokens issued on the old blockchain have also therefore forked, each from one token type into two, with each token type running on each separate blockchain.

How to overcome these problems?

Proposal

Our proposal, so as to reach as close as possible towards the requirement of extreme configurability, is to make every option that's currently available in the source code — every switch statement related to a blockchain — configurable as an option in JSON.

And then moving forward, whenever any new option is added to the source code (such as a new block format or signing algorithm), it should also be added as a new configuration option in a subsequent version of the JSON schema. In this way, any new blockchain that uses a known signing algorithm, or a known block format, etc., can be added to the wallet simply by adding the appropriate JSON data file.

Let's get right into the JSON schemas…

List of JSON datafiles and schemas described in this document:

address_style.json
address_style.schema.json
bip44_type.json
bip44_type.schema.json
blockchain.schema.json
blockchain_defs.schema.json
blockchain_type.json
blockchain_type.schema.json
contact_defs.json
contact_defs.schema.json
credential.schema.json
credential_defs.json
credential_defs.schema.json
defs.schema.json
key_defs.json
key_defs.schema.json
nym.schema.json
nym_defs.json
nym_defs.schema.json
off_chain_unit.schema.json
p2p_protocol_type.json
p2p_protocol_type.schema.json
p2p_service.json
p2p_service.schema.json
payment_code.schema.json
signature.schema.json
signature_defs.json
signature_defs.schema.json
token.schema.json
token_type.json
token_type.schema.json
unit_defs.schema.json
unit_type.json

Basic Definitions (defs.schema.json)

This file contains a few basic type definitions used throughout the rest of the JSON schemas:

defs.schema.json

```
{
    "description": "Basic definitions used in opentxs",
    "defs": {
        "int8": {
            "type": "integer",
            "minimum": -128,
            "maximum": 127
        },
        "int16": {
            "type": "integer",
            "minimum": -32768,
            "maximum": 32767
        },
        "int32": {
            "type": "integer",
            "minimum": -2147483648,
            "maximum": 2147483647
        },
        "int64": {
            "type": "integer",
            "minimum": -9223372036854775808,
            "maximum": 9223372036854775807
        },
        "uint8": {
            "type": "integer",
            "minimum": 0,
            "maximum": 255
        },
        "uint16": {
            "type": "integer",
            "minimum": 0,
            "maximum": 65535
        },
        "uint32": {
            "type": "integer",
            "minimum": 0,
```

```
            "maximum": 4294967295
        },
        "uint64": {
            "type": "integer",
            "minimum": 0,
            "maximum": 18446744073709551615
        },
        "mapStringString": {
            "type": "object",
            "patternProperties": {
                ".{1,}": {
                    "type": "string"
                }
            }
        },
        "ArrayOfStrings": {
            "type": "array",
            "items": {
                "type": "string"
            }
        },
        "FilePath": {
            "type": "string"
        }
    }
}
```

Address Style (Data File)

Below is the address_style.json data file, containing a list of the blockchain address styles supported by the wallet which are configurable for each blockchain.

These address styles depend on corresponding compiled code, and thus a recompile is necessary to add new address styles. As a result, this file is signed to prevent accidental edits.

address_style.json

```
{
    "version": 1,
    "address_style_map": {
        "P2PKH": 1,
        "P2SH": 2,
        "P2WPKH": 3,
        "P2WSH": 4,
        "P2TR": 5
    },
    "key_id": {
     "asd876asd766asd"
    },
    "signature": {
     "kjhasdf987q345kjh3q4kj5hadfs8yaskjakjhsfdi8uy34kjkjafdkjhasd"
    }
}
```

Address Style (Schema)

Below is the schema for the above address_style.json data file:

address_style.schema.json

```
{
    "description": "Address style related definitions used in opentxs",
    "type": "object",
    "properties": {
      "version": { "$ref": "defs.schema.json#/defs/uint32" },
      "address_style_map": { "$ref": "#/$defs/mapStringAddressStyle" },
      "key_id": { "string" },
      "signature": { "string" }
    },
    "required": ["version", "address_style_map", "key_id", "signature"],
    "additionalProperties": false,
    "$defs": {
        "AddressStyle": {
            "type": "integer",
            "minimum": 1,
            "maximum": 5
        },
        "mapStringAddressStyle": {
            "type": "object",
            "patternProperties": {
                ".{1,}": { "$ref": "#/$defs/AddressStyle" }
            }
        }
    }
}
```

Basic Definitions Related to Blockchains

Here are a few type definitions subsequently used in the main blockchain.schema.json (which follows this file):

blockchain_defs.schema.json

```
{
    "description": "Building blocks for blockchain-related definitions used in
opentxs",
    "$defs": {
        "ScriptMap": {
            "type": "array",
            "items": {
                "type": "object",
                "properties": {
                    "key": { "$ref": "address_style.schema.json#/$defs/
AddressStyle" },
                    "value": "boolean"
                }
            }
        },
        "StylePref": {
            "type": "array",
            "items": {
                "type": "object",
                "properties": {
                    "key": { "$ref": "address_style.schema.json#/$defs/
AddressStyle" },
                    "value": "string"
                }
            }
        },
        "BlockHash": { "type": "string" },
        "BlockHeight": { "$ref": "defs.schema.json#/defs/int64" },
        "FilterType": { "$ref": "defs.schema.json#/defs/uint32" },
        "FilterHeader": { "type": "string" },
        "Checkpoint": {
          "type": "object",
          "properties": {
            "height": { "$ref": "#/$defs/BlockHeight" },
            "block_hash": { "$ref": "#/$defs/BlockHash" },
            "previous_block_hash": { "$ref": "#/$defs/BlockHash" },
```

```
            "filter_header": { "$ref": "#/$defs/FilterHeader" }
          },
          "required": ["height", "block_hash", "previous_block_hash",
"filter_header"],
          "additionalProperties": false
        },
        "P2pProtocolVersion": { "type": "integer" },
        "BlockHeaderType" : {
            "type": "string",
            "enum": ["bitcoin", "ethereum", "casper"]
        },
        "TransactionBuilderType" : {
            "type": "string",
            "enum": ["bitcoin", "ethereum", "casper"]
        }
    }
}
```

Blockchain (Schema)

Here is the main schema for blockchain data. (Notes to follow).

blockchain.schema.json

```
{
  "$id": "https://matterfi.com/schemas/blockchain",
  "description": "Blockchain schema used in opentxs",
  "type": "object",
  "properties": {
    "version": { "$ref": "defs.schema.json#/defs/uint32" },
    "blockchain_type": { "$ref": "blockchain_type.schema.json#/defs/
BlockchainType" },
    "block_header_type": { "$ref": "blockchain_defs.schema.json#/defs/
BlockHeaderType" },
    "transaction_builder_type": { "$ref": "blockchain_defs.schema.json#/defs/
TransactionBuilderType" },
    "supported": { "boolean" },
    "testnet": { "boolean" },
    "item_type": { "$ref": "unit_defs.schema.json#/defs/UnitType" },
    "bip44": { "$ref": "bip44_type.schema.json#/defs/Bip44Type" },
    "bits": { "$ref": "defs.schema.json#/defs/int32" },
    "genesis_header_hex": { "string" },
    "genesis_hash_hex": { "string" },
    "genesis_block_hex": { "string" },
    "checkpoint": { "$ref": "blockchain_defs.schema.json#/defs/Checkpoint" },
    "default_filter_type": { "$ref": "blockchain_defs.schema.json#/defs/
FilterType" },
    "p2p_protocol": { "$ref": "p2p_protocol_type.schema.json#/defs/
P2pProtocolType" },
    "p2p_protocol_version": { "$ref": "blockchain_defs.schema.json#/defs/
P2pProtocolVersion" },
    "p2p_magic_bits": { "$ref": "defs.schema.json#/defs/uint32" },
```

```
    "default_port": { "$ref": "defs.schema.json#/defs/uint16" },
    "dns_seeds": { "$ref": "defs.schema.json#/defs/ArrayOfStrings" },
    "default_fee_rate": { "integer" },
    "block_download_batch": { "integer" },
    "maturation_interval": { "$ref": "blockchain_defs.schema.json#/defs/
BlockHeight" },
    "cfilter_element_count_estimate": { "integer" },
    "block_hashtype": { "string" },
    "filter_hashtype": { "string" },
    "p2p_message_hashtype": { "string" },
    "proof_of_work_hashtype": { "string" },
    "pubkey_hashtype": { "string" },
    "script_hashtype": { "string" },
    "script_segwit_hashtype": { "string" },
    "transaction_hashtype": { "string" },
    "bip158_filter_types": { "$ref": "#/defs/Bip158FilterTypeArray" },
    "filters": { "$ref": "#/defs/FilterDataArray" },
    "services" : { "$ref": "#/defs/mapP2pServiceByBit" },
    "oneOf": [
        { "bitcoin_params": { "$ref": "#/defs/BitcoinParams" } },
        { "ethereum_params": { "$ref": "#/defs/EthereumParams" } },
        { "casper_params": { "$ref": "#/defs/CasperParams" } }
    ],
    "oneOf": [
        {
            "initial_state" : {
                "description": "Properties for a chain that starts at genesis 0",
                "type": "object",
                "properties": {
                    "genesis_block_height": { "type": "integer", "enum" : [0] },
                },
                "required": ["genesis_block_height"],
                "additionalProperties": false
            }
```

```
        },
        {
            "continuing_state" : {
                "description": "Properties for a chain that has forked",
                "type": "object",
                "properties": {
                    "previous_blockchain_type": { "$ref":
"blockchain_type.schema.json#/defs/BlockchainType" },
                    "continuing_block_height": { "$ref":
"blockchain_defs.schema.json#/defs/BlockHeight" },
                    "continuing_header_hex": { "string" },
                    "continuing_hash_hex": { "string" },
                    "continuing_block_hex": { "string" }
                },
                "required": ["previous_blockchain_type", "continuing_block_height",
“continuing_header_hex”, “continuing_hash_hex”, “continuing_block_hex”],
                "additionalProperties": false
            }
        }
    ]
  },
  "required": ["version", "blockchain_type", "block_header_type",
      "transaction_builder_type", "supported", "testnet", "item_type", "bip44",
      "bits", "genesis_header_hex", "genesis_hash_hex", "genesis_block_hex",
      "checkpoint", "default_filter_type", "p2p_protocol", "p2p_protocol_version",
      "p2p_magic_bits", "default_port", "dns_seeds", "default_fee_rate",
      "block_download_batch", "maturation_interval",
"cfilter_element_count_estimate",
      "block_hashtype", "filter_hashtype", "p2p_message_hashtype",
      "proof_of_work_hashtype", "pubkey_hashtype", "script_hashtype",
      "script_segwit_hashtype", "transaction_hashtype", "bip158_filter_types",
      "filters", "services"],
  "oneOf": [
      { "required": [ "bitcoin_params" ] },
```

```
      { "required": [ "ethereum_params" ] },
      { "required": [ "casper_params" ] }
  ],
  "oneOf": [
      { "required": [ "initial_state" ] },
      { "required": [ "continuing_state" ] }
  ],
  "additionalProperties": false,
  "$defs": {
      "BitcoinParams": {
          "description": "Properties specific to Bitcoin-style chains. Note: Still
missing SigHash flags",
          "type": "object",
          "properties": {
              "version": { "$ref": "defs.schema.json#/defs/uint32" },
              "segwit": { "boolean" },
              "segwit_scale_factor": { "integer" },
              "script_map": { "$ref": "blockchain_defs.schema.json#/defs/
ScriptMap" },
              "styles": { "$ref": "blockchain_defs.schema.json#/defs/StylePref" },
              "block_type" : {
                  "type": "string",
                  "enum": ["normal", "pkt"]
              },
              "input_signature_type" : {
                  "type": "string",
                  "enum": ["bch", "segwit"]
              }
          },
          "required": ["version", "segwit", "segwit_scale_factor", "script_map",
              "styles", "block_type"],
          "additionalProperties": false
      },
      "EthereumParams": {
```

```
            "description": "Properties specific to Ethereum-style chains",
            "type": "object",
            "properties": {
                "version": { "$ref": "defs.schema.json#/defs/uint32" }
            },
            "required": ["version"],
            "additionalProperties": false
        },
        "CasperParams": {
            "description": "Properties specific to Casper-style chains",
            "type": "object",
            "properties": {
                "version": { "$ref": "defs.schema.json#/defs/uint32" }
            },
            "required": ["version"],
            "additionalProperties": false
        },
        "Bip158FilterTypeArray": {
            "description": "Bip-158 filter types specific to a given blockchain",
            "type": "array",
            "items": {
                "type": "object",
                "properties": {
                    "filter_type": { "$ref": "blockchain_defs.schema.json#/defs/
FilterType" },
                    "filter_value": { "$ref": "defs.schema.json#/defs/uint8" }
                },
                "additionalProperties": false
            }
        },
        "FilterDataArray": {
            "description": "Filter data specific to a given blockchain",
            "type": "array",
            "items": {
```

```
            "type": "object",
            "properties": {
                "filter_type": { "$ref": "blockchain_defs.schema.json#/defs/
FilterType" },
                "filter_value": { "string" }
            },
            "additionalProperties": false
        }
    },
    "mapP2pServiceByBit": {
        "type": "object",
        "description" : "key is p2p service bit, value is p2p service",
        "patternProperties": {
            ".{1,}": { "$ref": "p2p_service.schema.json#/$defs/P2pService" }
        }
    }
  }
}
```

Notes on blockchain.schema.json

Below are a few relevant notes for the properties of the blockchain schema (above):

- “version” for updating the blockchain schema.
- “blockchain_type” is a unique identifier for each blockchain. Here is the current datafile for the known blockchain types (**blockchain_type.json**):

```
{
    "version": 1,
    "blockchain_type_map": {
        "Bitcoin": 1,
        "Bitcoin_testnet3": 2,
        "BitcoinCash": 3,
        "BitcoinCash_testnet3": 4,
        "Ethereum_frontier": 5,
        "Ethereum_ropsten": 6,
        "Litecoin": 7,
        "Litecoin_testnet4": 8,
        "PKT": 9,
        "PKT_testnet": 10,
        "BitcoinSV": 11,
        "BitcoinSV_testnet3": 12,
        "eCash": 13,
        "eCash_testnet3": 14
    }
}
```

- “block_header_type” as described in blockchain_defs.schema.json, the available types are currently ["bitcoin", "ethereum", "casper”].
- “transaction_builder_type” as described in blockchain_defs.schema.json, the available types are currently ["bitcoin", "ethereum", "casper”].

- “supported” a boolean describing whether the blockchain is currently supported.
- “testnet” a boolean, describes whether or not this blockchain is a testnet.
- “item_type” is a UnitType as defined in **unit_defs.schema.json**. (The datafile for the unit types is signed and will be described in the next section of this document). Meanwhile, here is the schema:

```
{
    "description": "Unit type related definitions used in opentxs",
    "type": "object",
    "properties": {
      "version": { "$ref": "defs.schema.json#/defs/uint32" },
      "unit_type_map": { "$ref": "#/$defs/mapStringUnitType" },
      "equity_type_map": { "$ref": "#/$defs/mapStringEquityType" },
      "key_id": { "string" },
      "signature": { "string" }
    },
    "required": ["version", "unit_type_map", “equity_type_map”, ”key_id",
"signature"],
    "additionalProperties": false,
    "$defs": {
        "UnitType": { "type": "integer", "minimum" : 1, "maximum" : 300 },
        "mapStringUnitType": {
            "type": "object",
            "patternProperties": {
                ".{1,}": { "$ref": "#/$defs/UnitType" }
            }
        },
        "EquityType": { "type": "integer", "minimum" : 0, "maximum" : 1 },
        "mapStringEquityType": {
            "type": "object",
            "patternProperties": {
                ".{1,}": { "$ref": "#/$defs/EquityType" }
            }
```

```
            }
        }
    }
```

- "bip44" is a Bip44Type from **bip44_type.json**:

```
{
    "version": 1,
    "bip44_type_map": {
        "BITCOIN" : 0,
        "TESTNET" : 1,
        "LITECOIN" : 2,
        "DOGECOIN" : 3,
        "REDDCOIN" : 4,
        "DASH" : 5,
        "PEERCOIN" : 6,
        "NAMECOIN" : 7,
        "FEATHERCOIN" : 8,
        "COUNTERPARTY" : 9,
        "BLACKCOIN" : 10,
        "ETHER" : 60,
        "BITCOINCASH" : 145,
        "BITCOINSV" : 236,
        "PKT" : 390,
        "ECASH" : 899
    }
}
```

- Here is the corresponding definition, from **bip44_type.schema.json**:

```
    {
        "description": "BIP-44 related definitions used in opentxs",
        "type": "object",
        "properties": {
          "version": { "$ref": "defs.schema.json#/defs/uint32" },
          "bip44_type_map": { "$ref": "#/$defs/mapStringBip44Type" }
        },
        "required": ["version", "bip44_type_map"],
```

```
    "additionalProperties": false,
    "$defs": {
        "Bip44Type": {
            “oneOf”: [
              {
                "type": "integer",
                "minimum": 0,
                "maximum": 10
              },
              {
                "type": "integer",
                "enum": [60, 145, 236, 390, 899]
              }
            ]
        },
        "mapStringBip44Type": {
            "type": "object",
            "patternProperties": {
                ".{1,}": { "$ref": "#/$defs/Bip44Type" }
            }
        }
    }
}
```

- "bits" as described in https://btcinformation.org/en/developer-reference#target-nbits
- "genesis_header_hex" a hexadecimal representation containing the genesis header for this blockchain.
- "genesis_hash_hex" a hexadecimal representation containing the hash of the genesis header for this blockchain.
- "genesis_block_hex" a hexadecimal representation containing the genesis block for this blockchain.
- "checkpoint" the most recent relevant checkpoint for this blockchain.

- "default_filter_type" the default filter type for this blockchain.
- "p2p_protocol" the p2p protocol used by this blockchain. Available types are listed in this file:

p2p_protocol_type.json

```
{
    "version": 1,
    "p2p_protocol_type_map": {
        "opentxs": 0,
        "bitcoin": 1,
        "ethereum": 2
    },
    "key_id": {
        "asd876asd766asd"
    },
    "signature": {
        "kjhasdf987q345kjh3q4kj5hadfs8yaskjakjhsfdi8uy34kjkjafdkjhasd"
    }
}
```

- The schema for the above p2p protocol type list is defined in **p2p_protocol_type.schema.json**:

```
{
    "description": "P2p protocol type related definitions used in opentxs",
    "type": "object",
    "properties": {
      "version": { "$ref": "defs.schema.json#/defs/uint32" },
      "p2p_protocol_type_map": { "$ref": "#/$defs/
mapStringP2pProtocolType" },
      "key_id": { "string" },
      "signature": { "string" }
    },
    "required": ["version", "p2p_protocol_type_map", "key_id", "signature"],
    "additionalProperties": false,
    "$defs": {
        "P2pProtocolType": {
            "type": "integer",
            "minimum": 0,
            "maximum": 2
        },
        "mapStringP2pProtocolType": {
            "type": "object",
            "patternProperties": {
                ".{1,}": { "$ref": "#/$defs/P2pProtocolType" }
            }
```

```
        }
    }
}
```

- "p2p_protocol_version" the version used, in the code, of the p2p protocol used by this blockchain.
- "p2p_magic_bits" the magic bits used to determine the beginning of a protocol message for the p2p protocol used by this blockchain.
- "default_port" the default port used by the p2p protocol for this blockchain.
- "dns_seeds" the DNS seeds used by the p2p protocol for this blockchain.
- "default_fee_rate" the fee rate is calculated from the mempool, and queried from peers when supported by the protocol. Until then, the default_fee_rate is configured here in the JSON datafile. This is measured in Satoshis per 1000 bytes. It's something we have to check the reference node implementation for, to see what fee rates will be relayed by default. It can change from time to time though so it needs to be periodically re-checked.
- "block_download_batch" the number of blocks in a given download batch for this blockchain. This is a safety feature we use to ensure that we don't try to load too many blocks in memory as we download them.
- "maturation_interval" the number of blocks between when new coins are mined and when they become spendable. This is usually 100 for every blockchain, and it's part of a blockchain's consensus definition.
- "block_hashtype" the hash algorithm used for the block hash.
- "cfilter_element_count_estimate" the estimate of the number of elements used in a typical filter for this blockchain. Used to seed the adaptive scan interval algorithm. It's something we have to derive ourselves and its purpose is to keep the algorithm from attempting to scan too many cfilters on its first run, before it has collected any data regarding the average cfilter size.
- "filter_hashtype" the hash algorithm used for the filter hash.
- "p2p_message_hashtype" the hash algorithm used for the p2p message hash.
- "proof_of_work_hashtype" the hash algorithm used for proof-of-work for this

blockchain.

- "pubkey_hashtype" the hash algorithm used for the pub key hash for this blockchain.
- "script_hashtype" the hash algorithm used for scripts for this blockchain.
- "script_segwit_hashtype" the hash algorithm used for segwit scripts for this blockchain.
- "transaction_hashtype" the hash algorithm used for transaction hashes for this blockchain.
- "bip158_filter_types" the bip158 filter types supported for this blockchain.
- "filters" an array of [filter type, filter data] for this blockchain.
- "services" a map of p2p services for a given blockchain. The key is the p2p service bit, and the value is the p2p service. The schema for the p2p services is found in this file:

p2p_service.schema.json

```
{
    "description": "P2p service related definitions used in opentxs",
    "type": "object",
    "properties": {
      "version": { "$ref": "defs.schema.json#/defs/uint32" },
      "p2p_service_map": { "$ref": "#/$defs/mapStringP2pService" },
      "p2p_service_bit_map": { "$ref": "#/$defs/mapStringP2pServiceBits" },
      "key_id": { "string" },
      "signature": { "string" }
    },
    "required": ["version", "p2p_service_map", "p2p_service_bit_map",
        "key_id", "signature"],
    "additionalProperties": false,
    "$defs": {
        "P2pService":
        {
            "type": "integer",
            "minimum": 0,
            "maximum": 13
        },
        "P2pServiceBit":
        {
            "type": "integer",
            "minimum": 0,
```

```
            "maximum": 64
        },
        "mapStringP2pServiceBits": {
            "type": "object",
            "patternProperties": {
                ".{1,}": { "$ref": "#/$defs/P2pServiceBit" }
            }
        },
        "mapStringP2pService": {
            "type": "object",
            "patternProperties": {
                ".{1,}": { "$ref": "#/$defs/P2pService" }
            }
        }
    }
}
```

- The datafile is located here:

p2p_service.json

```
{
    "version": 1,
    "p2p_service_map": {
        "None": 0,
        "Avalanche": 1,
        "BitcoinCash": 2,
        "Bloom": 3,
        "CompactFilters": 4,
        "Graphene": 5,
        "Limited": 6,
        "Network": 7,
        "Segwit2X": 8,
        "UTXO": 9,
        "WeakBlocks": 10,
        "Witness": 11,
        "XThin": 12,
        "XThinner": 13
    },
    "p2p_service_bit_map": {
        "None": 0,
        "Bit1" : 1,
        "Bit2" : 2,
        "Bit3" : 3,
        "Bit4" : 4,
        "Bit5" : 5,
        "Bit6" : 6,
        "Bit7" : 7,
```

```
"Bit8" : 8,
"Bit9" : 9,
"Bit10" : 10,
"Bit11" : 11,
"Bit12" : 12,
"Bit13" : 13,
"Bit14" : 14,
"Bit15" : 15,
"Bit16" : 16,
"Bit17" : 17,
"Bit18" : 18,
"Bit19" : 19,
"Bit20" : 20,
"Bit21" : 21,
"Bit22" : 22,
"Bit23" : 23,
"Bit24" : 24,
"Bit25" : 25,
"Bit26" : 26,
"Bit27" : 27,
"Bit28" : 28,
"Bit29" : 29,
"Bit30" : 30,
"Bit31" : 31,
"Bit32" : 32,
"Bit33" : 33,
"Bit34" : 34,
"Bit35" : 35,
"Bit36" : 36,
"Bit37" : 37,
"Bit38" : 38,
"Bit39" : 39,
"Bit40" : 40,
"Bit41" : 41,
"Bit42" : 42,
"Bit43" : 43,
"Bit44" : 44,
"Bit45" : 45,
"Bit46" : 46,
"Bit47" : 47,
"Bit48" : 48,
"Bit49" : 49,
"Bit50" : 50,
"Bit51" : 51,
"Bit52" : 52,
"Bit53" : 53,
```

```
        "Bit54" : 54,
        "Bit55" : 55,
        "Bit56" : 56,
        "Bit57" : 57,
        "Bit58" : 58,
        "Bit59" : 59,
        "Bit60" : 60,
        "Bit61" : 61,
        "Bit62" : 62,
        "Bit63" : 63,
        "Bit64" : 64
    };
    "key_id": {
      "asd876asd766asd"
    },
    "signature": {
      "kjhasdf987q345kjh3q4kj5hadfs8yaskjakjhsfdi8uy34kjkjafdkjhasd"
    }
}
```

- "oneOf": ["bitcoin_params", "ethereum_params", "casper_params"] A data structure containing unique information for this blockchain, depending on whether it's a bitcoin, ethereum, or casper-style chains. (Subsequent versions of this schema will include every possible type of chain supported by the wallet).
- "oneOf": ["initial_state", "continuing_state"] If a blockchain has continued operating with a change in rules at a given block height, or if it has forked (meaning both sets of rules are now operating as 2 separate blockchains), then a separate blockchain entry must be added to the list. One of them contains the initial state from the genesis block, and the next entry contains the continuing state from a given checkpoint.

Unit Types

To understand unit types, consider a few examples with a USD unit type:

- There could be an off-chain currency for a **USD unit type** from a specific issuer.
- There could also be an on-blockchain token for a **USD unit type** from a different issuer.
- There could also be an off-chain currency that represents server deposits of the above USD on-blockchain token. Units of that off-chain currency also have a **USD unit type**.

Let's also consider a few examples with the BTC unit type:

- There is a blockchain (Bitcoin) with a native coin of **BTC unit type**.
- There could also be ERC-20 tokens circulating on the Ethereum chain, which represent "wrapped BTC" and thus have a **BTC unit type**.
- There could be an off-chain currency that represents server deposits of the above ERC-20 token, again with a **BTC unit type**.

NOTE: It's crucial that all wallets agree on which unit type is which. If one wallet believes that unit type 1 is BTC (which it is) while another wallet believes that unit type 1 is ETH (which it is not) then those wallets are no longer able to interact meaningfully regarding those unit types.

As a result, the opentxs project maintains a canonical list of unit types, which includes both ISO 4217 currency codes as well as blockchain currencies. (There is no other universal list of currencies, and so the opentxs project must maintain our own).

While the opentxs project does maintain a canonical list, the requirements of this DevXDao milestone have prompted us to make this list available as a signed JSON

file, rather than compiling it directly into the binary, so that the maintainers of the project can update it without having to recompile the wallet.

Here is version 1 of the unit type JSON data file:

unit_type.json

```
{
    "version": 1,
    "unit_type_map": {
        "Btc" : 1,
        "Bch" : 2,
        "Eth" : 3,
        "Xrp" : 4,
        "Ltc" : 5,
        "Dao" : 6,
        "Xem" : 7,
        "Dash" : 8,
        "Maid" : 9,
        "Lsk" : 10,
        "Doge" : 11,
        "Dgd" : 12,
        "Xmr" : 13,
        "Waves" : 14,
        "Nxt" : 15,
        "Sc" : 16,
        "Steem" : 17,
        "Amp" : 18,
        "Xlm" : 19,
        "Fct" : 20,
        "Bts" : 21,
        "Usd" : 22,
        "Eur" : 23,
        "Gbp" : 24,
        "Inr" : 25,
```

```
"Aud" : 26,
"Cad" : 27,
"Sgd" : 28,
"Chf" : 29,
"Myr" : 30,
"Jpy" : 31,
"Cny" : 32,
"Nzd" : 33,
"Thb" : 34,
"Huf" : 35,
"Aed" : 36,
"Hkd" : 37,
"Mxn" : 38,
"Zar" : 39,
"Php" : 40,
"Sek" : 41,
"Pkt" : 42,
"Tnbtc" : 43,
"Tnbch" : 44,
"Tnxrp" : 45,
"Tnltx" : 46,
"Tnxem" : 47,
"Tndash" : 48,
"Tnmaid" : 49,
"Tnlsk" : 50,
"Tndoge" : 51,
"Tnxmr" : 52,
"Tnwaves" : 53,
"Tnnxt" : 54,
"Tnsc" : 55,
"Tnsteem" : 56,
"Tnpkt" : 57,
"Ethereum_olympic" : 58,
"Ethereum_classic" : 59,
```

```
"Ethereum_expanse" : 60,
"Ethereum_morden" : 61,
"Ethereum_ropsten" : 62,
"Ethereum_rinkeby" : 63,
"Ethereum_kovan" : 64,
"Ethereum_sokol" : 65,
"Ethereum_poa" : 66,
"Regtest" : 67,
"Bnb" : 68,
"Sol" : 69,
"Usdt" : 70,
"Ada" : 71,
"Dot" : 72,
"Usdc" : 73,
"Shib" : 74,
"Luna" : 75,
"Avax" : 76,
"Uni" : 77,
"Link" : 78,
"Wbtc" : 79,
"Busd" : 80,
"MatiC" : 81,
"Algo" : 82,
"Vet" : 83,
"Axs" : 84,
"Icp" : 85,
"Cro" : 86,
"Atom" : 87,
"Theta" : 88,
"Fil" : 89,
"Trx" : 90,
"Ftt" : 91,
"Etc" : 92,
"Ftm" : 93,
```

```
"Dai" : 94,
"Btcb" : 95,
"Egld" : 96,
"Hbar" : 97,
"Xtz" : 98,
"Mana" : 99,
"Near" : 100,
"Grt" : 101,
"Cake" : 102,
"Eos" : 103,
"Flow" : 104,
"Aave" : 105,
"Klay" : 106,
"Ksm" : 107,
"Xec" : 108,
"Miota" : 109,
"Hnt" : 110,
"Rune" : 111,
"Bsv" : 112,
"Leo" : 113,
"Neo" : 114,
"One" : 115,
"Qnt" : 116,
"Ust" : 117,
"Mkr" : 118,
"Enj" : 119,
"Chz" : 120,
"Ar" : 121,
"Stx" : 122,
"Btt" : 123,
"Hot" : 124,
"Sand" : 125,
"Omg" : 126,
"Celo" : 127,
```

```
"Zec" : 128,
"Comp" : 129,
"Tfuel" : 130,
"Kda" : 131,
"Lrc" : 132,
"Qtum" : 133,
"Crv" : 134,
"Ht" : 135,
"Nexo" : 136,
"Sushi" : 137,
"Kcs" : 138,
"Bat" : 139,
"Okb" : 140,
"Dcr" : 141,
"Icx" : 142,
"Rvn" : 143,
"Scrt" : 144,
"Rev" : 145,
"Audio" : 146,
"Zil" : 147,
"Tusd" : 148,
"Yfi" : 149,
"Mina" : 150,
"Perp" : 151,
"Xdc" : 152,
"Tel" : 153,
"Snx" : 154,
"Btg" : 155,
"Afn" : 156,
"All" : 157,
"Amd" : 158,
"Ang" : 159,
"Aoa" : 160,
"Ars" : 161,
```

```
"Awg" : 162,
"Azn" : 163,
"Bam" : 164,
"Bbd" : 165,
"Bdt" : 166,
"Bgn" : 167,
"Bhd" : 168,
"Bif" : 169,
"Bmd" : 170,
"Bnd" : 171,
"Bob" : 172,
"Brl" : 173,
"Bsd" : 174,
"Btn" : 175,
"Bwp" : 176,
"Byn" : 177,
"Bzd" : 178,
"Cdf" : 179,
"Clp" : 180,
"Cop" : 181,
"Crc" : 182,
"Cuc" : 183,
"Cup" : 184,
"Cve" : 185,
"Czk" : 186,
"Djf" : 187,
"Dkk" : 188,
"Dop" : 189,
"Dzd" : 190,
"Egp" : 191,
"Ern" : 192,
"Etb" : 193,
"Fjd" : 194,
"Fkp" : 195,
```

```
"Gel" : 196,
"Ggp" : 197,
"Ghs" : 198,
"Gip" : 199,
"Gmd" : 200,
"Gnf" : 201,
"Gtq" : 202,
"Gyd" : 203,
"Hnl" : 204,
"Hrk" : 205,
"Htg" : 206,
"Idr" : 207,
"Ils" : 208,
"Imp" : 209,
"Iqd" : 210,
"Irr" : 211,
"Isk" : 212,
"Jep" : 213,
"Jmd" : 214,
"Jod" : 215,
"Kes" : 216,
"Kgs" : 217,
"Khr" : 218,
"Kmf" : 219,
"Kpw" : 220,
"Krw" : 221,
"Kwd" : 222,
"Kyd" : 223,
"Kzt" : 224,
"Lak" : 225,
"Lbp" : 226,
"Lkr" : 227,
"Lrd" : 228,
"Lsl" : 229,
```

```
"Lyd" : 230,
"Mad" : 231,
"Mdl" : 232,
"Mga" : 233,
"Mkd" : 234,
"Mmk" : 235,
"Mnt" : 236,
"Mop" : 237,
"Mru" : 238,
"Mur" : 239,
"Mvr" : 240,
"Mwk" : 241,
"Mzn" : 242,
"Nad" : 243,
"Ngn" : 244,
"Nio" : 245,
"Nok" : 246,
"Npr" : 247,
"Omr" : 248,
"Pab" : 249,
"Pen" : 250,
"Pgk" : 251,
"Pkr" : 252,
"Pln" : 253,
"Pyg" : 254,
"Qar" : 255,
"Ron" : 256,
"Rsd" : 257,
"Rub" : 258,
"Rwf" : 259,
"Sar" : 260,
"Sbd" : 261,
"Scr" : 262,
"Sdg" : 263,
```

```
"Shp" : 264,
"Sll" : 265,
"Sos" : 266,
"Spl" : 267,
"Srd" : 268,
"Stn" : 269,
"Svc" : 270,
"Syp" : 271,
"Szl" : 272,
"Tjs" : 273,
"Tmt" : 274,
"Tnd" : 275,
"Top" : 276,
"Try" : 277,
"Ttd" : 278,
"Tvd" : 279,
"Twd" : 280,
"Tzs" : 281,
"Uah" : 282,
"Ugx" : 283,
"Uyu" : 284,
"Uzs" : 285,
"Vef" : 286,
"Vnd" : 287,
"Vuv" : 288,
"Wst" : 289,
"Xaf" : 290,
"Xcd" : 291,
"Xdr" : 292,
"Xof" : 293,
"Xpf" : 294,
"Yer" : 295,
"Zmw" : 296,
"Zwd" : 297,
```

```
        "Custom" : 298,
        "Tnbsv" : 299,
        "TnXec" : 300
    },
    "equity_type_map": {
        "EQUITYTYPE_ERROR" : 0,
        "EQUITYTYPE_SHARES" : 1
    },
    "key_id": {
     "asd876asd766asd"
    },
    "signature": {
     "kjhasdf987q345kjh3q4kj5hadfs8yaskjakjhsfdi8uy34kjkjafdkjhasd"
    }
}
```

On-Chain Tokens

Tokens are issued as units that circulate on-blockchain. Known protocols for issuing and transacting tokens include SLP (for Bitcoin-style chains) and ERC-20 (for Ethereum-style chains).

Supported token types are listed in this file:

token_type.json

```
{
    "version": 1,
    "token_type_map": {
        "ERC20": 0,
        "SLP": 1
    },
    "key_id": {
      "asd876asd766asd"
    },
    "signature": {
     "kjhasdf987q345kjh3q4kj5hadfs8yaskjakjhsfdi8uy34kjkjafdkjhasd"
    }
}
```

Here is the schema for the above file:

token_type.schema.json

```
{
    "description": "Token type related definitions used in opentxs",
    "type": "object",
    "properties": {
      "version": { "$ref": "defs.schema.json#/defs/uint32" },
      "token_type_map": { "$ref": "#/$defs/mapStringTokenType" },
      "key_id": { "string" },
```

```
      "signature": { "string" }
    },
    "required": ["version", "token_type_map", "key_id", "signature"],
    "additionalProperties": false,
    "$defs": {
        "TokenType": {
            "type": "integer",
            "minimum": 0,
            "maximum": 1
        },
        "mapStringTokenType": {
            "type": "object",
            "patternProperties": {
                ".{1,}": { "$ref": "#/$defs/TokenType" }
            }
        }
    }
}
```

The below file contains the schema for adding new tokens to the wallet. (Notes to follow).

token.schema.json

```
{
  "$id": "https://matterfi.com/schemas/token",
  "description": "Token schema used in opentxs",
  "type": "object",
  "properties": {
    "version": { "$ref": "defs.schema.json#/defs/uint32" },
    "token_id": { "string" },
    "blockchain_type": { "$ref": "blockchain_type.schema.json#/defs/
BlockchainType" },
    "token_type": { "$ref": "token_type.schema.json#/defs/TokenType" },
    "unit_type": { "$ref": "unit_defs.schema.json#/defs/UnitType" },
```

```
  "genesis_txn_header_hex": { "string" },
  "genesis_txn_hash_hex": { "string" },
  "genesis_txn_block_hex": { "string" },
  "oneOf": [
      { "slp_params": { "$ref": "#/defs/SlpParams" } },
      { "erc20_params": { "$ref": "#/defs/Erc20Params" } }
   ]
},
"required": ["version", "token_id", "blockchain_type", "token_type",
    "genesis_txn_header_hex", "genesis_txn_hash_hex", "genesis_txn_block_hex"],
"oneOf": [
    { "required": [ "slp_params" ] },
    { "required": [ "erc20_params" ] }
],
"additionalProperties": false,
"$defs": {
    "SlpParams": {
        "description": "Properties specific to SLP-style tokens",
        "type": "object",
        "properties": {
            "version": { "$ref": "defs.schema.json#/defs/uint32" },
            "name": { "string" },
            "ticker_symbol": { "string" },
            "document_url": { "string" },
            "receiver_address": { "string" },
            "baton_address": { "string" }
        },
        "required": ["version", "receiver_address", "baton_address"],
        "additionalProperties": false
    },
    "Erc20Params": {
        "description": "Properties specific to ERC20-style tokens",
        "type": "object",
        "properties": {
```

```
            "version": { "$ref": "defs.schema.json#/defs/uint32" },
            "contract_hex": { "string" }
            "contract_hash_hex": { "string" }
            "contract_address": { "string" }
        },
        "required": ["version", "contract_hex", "contract_hash_hex"],
        "additionalProperties": false
    }
  }
}
```

Notes on token.schema.json

Below are a few relevant notes for the properties of the token schema (above):

- "version" for updating the token schema.
- "token_id" unique ID for each individual token.
- "blockchain_type" the blockchain where the token resides.
- "token_type" Currently a choice of ERC-20 or SLP. More coming.
- "unit_type" This is an optional property, since there may be thousands of ERC-20 token types of new or unknown unit types.
- "genesis_txn_header_hex" This is the header for the genesis transaction for this token.
- "genesis_txn_hash_hex" This is the hash of the genesis transaction for this token.
- "genesis_txn_block_hex" This is the contents of the genesis block for this transaction.
- "oneOf": ["slp_params", "erc20_params"] This value contains properties that are specific to the token type. Current types are ERC-20 and SLP.

Off-Chain Currencies (Schema)

Here is the main schema for off-chain currrencies. (Related schemas to follow).

off_chain_unit.schema.json

```
{
  "$id": "https://matterfi.com/schemas/off_chain_unit",
  "description": "Off-chain unit schema used in opentxs",
  "type": "object",
  "properties": {
    "version": { "$ref": "defs.schema.json#/defs/uint32" },
    "unit_definition_id": { "string" },
    "oneOf": [
        { "unit_type": { "$ref": "unit_defs.schema.json#/defs/UnitType" } },
        { "blockchain_type": {
            "$ref": "blockchain_type.schema.json#/defs/BlockchainType" } },
        { "token_id": { "$ref": "token_type.schema.json#/defs/
TokenType" } },
        { "currency_params": { "$ref": "#/defs/CurrencyParams" } },
        { "equity_params": { "$ref": "#/defs/EquityParams" } }
     ],
    "name": { "string" },
    "terms": { "string" },
    "redemption_increment": { "$ref": "defs.schema.json#/defs/uint64" },
    "issuer_nym": { "$ref": "nym.schema.json#" },
    "signature": { "string" }
  },
  "required": ["version", "unit_definition_id", "issuer_nym", "signature"],
  "oneOf": [
      { "required": [ "unit_type" ] },
      { "required": [ "blockchain_type" ] },
```

```
        { "required": [ "token_id" ] },
        { "required": [ "currency_params" ] },
        { "required": [ "equity_params" ] }
    ],
    "additionalProperties": false,
    "$defs": {
        "ScaleRatio": {
            "type": "object",
            "properties": {
                "version": { "$ref": "defs.schema.json#/defs/uint32" },
                "base": { "$ref": "defs.schema.json#/defs/uint32" },
                "power": { "$ref": "defs.schema.json#/defs/int32" }
            },
            "required": ["version", "base", "power"],
            "additionalProperties": false
        },
        "DisplayScale": {
            "type": "object",
            "properties": {
                "version": { "$ref": "defs.schema.json#/defs/uint32" },
                "scale_name": { "string" },
                "prefix": { "string" },
                "suffix": { "string" },
                "default_minimum_decimals": { "$ref": "defs.schema.json#/defs/
uint32" },
                "default_maximum_decimals": { "$ref": "defs.schema.json#/defs/
uint32" },
                "ratios": {
                    "type": "array",
                    "items": { "type": { "$ref": "#/defs/ScaleRatio" } }
                }
            },
            "required": ["version", "scale_name", "prefix", "suffix",
                "default_minimum_decimals", "default_maximum_decimals",
```

```
"ratios"],
          "additionalProperties": false
      },
      "CurrencyParams": {
          "description": "Properties specific to off-chain currencies",
          "type": "object",
          "properties": {
              "version": { "$ref": "defs.schema.json#/defs/uint32" },
              "unit_of_account": { "$ref": "contact_defs.schema.json#/$defs/
ContactItemType" },
              "short_name": { "string" },
              "scales": {
                  "type": "array",
                  "items": { "type": { "$ref": "#/defs/DisplayScale" } }
              }
          },
          "required": ["version", "unit_of_account", "short_name",
"scales"],
          "additionalProperties": false
      },
      "EquityParams": {
          "description": "Properties specific to off-chain equities
(shares)",
          "type": "object",
          "properties": {
              "version": { "$ref": "defs.schema.json#/defs/uint32" },
              "equity_type": { "$ref": "unit_defs.schema.json#/defs/
EquityType" }
          },
          "required": ["version", "equity_type"],
          "additionalProperties": false
      }
  }
}
```

(The following schemas are components to the one above).

Nym

The opentxs user schema used for defining the issuer nym for an off-chain currency.

nym.schema.json

```
{
  "$id": "https://matterfi.com/schemas/nym",
  "description": "Nym schema used in opentxs",
  "type": "object",
  "properties": {
    "version": { "$ref": "defs.schema.json#/defs/uint32" },
    "nym_id": { "string" },
    "nym_mode": { "$ref": "nym_defs.schema.json#/defs/NymMode" },
    "index": { "$ref": "defs.schema.json#/defs/uint32" },
    "revision": { "$ref": "defs.schema.json#/defs/uint64" },
    "nym_id_source": { "$ref": "nym_defs.schema.json#/defs/NymIDSource" },
    "active_credentials": { "$ref": "nym_defs.schema.json#/defs/AuthorityArray" },
    "revoked_credentials": { "$ref": "nym_defs.schema.json#/defs/AuthorityArray" },
    "signature": { "$ref": "signature_defs.schema.json#" }
  },
  "required": ["version", "nym_id", "nym_mode", "index", "revision",
      "nym_id_source", "active_credentials", "revoked_credentials", "signature"],
  "additionalProperties": false
}
```

nym_defs.schema.json

```
{
  "description": "Nym definitions used in opentxs",
  "type": "object",
  "properties": {
    "version": { "$ref": "defs.schema.json#/defs/uint32" },
```

```
    "nym_mode_map": { "$ref": "#/$defs/mapStringNymMode" },
    "source_type_map": { "$ref": "#/$defs/mapStringSourceType" },
    "authority_mode_map": { "$ref": "#/$defs/mapStringAuthorityMode" },
    "key_id": { "string" },
    "signature": { "string" }
  },
  "required": ["version", "nym_mode_map", "source_type_map", "key_id",
"signature"],
  "additionalProperties": false,
  "$defs": {
      "NymMode": { "type": "integer", "minimum" : 0, "maximum" : 2 },
      "mapStringNymMode": {
          "type": "object",
          "patternProperties": {
              ".{1,}": { "$ref": "#/$defs/NymMode" }
          }
      },
      "SourceType": { "type": "integer", "minimum" : 0, "maximum" : 2 },
      "mapStringSourceType": {
          "type": "object",
          "patternProperties": {
              ".{1,}": { "$ref": "#/$defs/SourceType" }
          }
      },
      "AuthorityMode": { "type": "integer", "minimum" : 0, "maximum" : 2 },
      "mapStringAuthorityMode": {
          "type": "object",
          "patternProperties": {
              ".{1,}": { "$ref": "#/$defs/AuthorityMode" }
          }
      },
      "PaymentCode": {
          "type": "object",
          "properties": {
```

```
            "version": { "$ref": "defs.schema.json#/defs/uint32" },
            "key": { "string" },
            "chaincode": { "string" },
            "bitmessage": { "boolean" },
            "bitmessage_version": { "$ref": "defs.schema.json#/defs/uint32" },
            "bitmessage_stream": { "$ref": "defs.schema.json#/defs/uint32" }
        },
        "required": ["version", "key", "chaincode", "bitmessage"],
        "additionalProperties": false
    },
    "NymIDSource": {
        "type": "object",
        "properties": {
            "version": { "$ref": "defs.schema.json#/defs/uint32" },
            "type": { "$ref": "#/defs/SourceType" },
            "oneOf": [
                { "source_key": { "$ref": "key_defs.schema.json#/defs/
AsymmetricKey" } },
                { "source_paymentcode": { "$ref": "#/defs/PaymentCode" } }
             ]
        },
        "required": ["version", "type"],
        "oneOf": [
            { "required": [ "source_key" ] },
            { "required": [ "source_paymentcode" ] }
        ],
        "additionalProperties": false
    },
    "CredentialArray": {
        "type": "array",
        "items": {
            "credential": { "$ref": "credential.schema.json#" }
        }
    },
```

```
      "Authority": {
          "description": "Todo: update 'required' list",
          "type": "object",
          "properties": {
              "version": { "$ref": "defs.schema.json#/defs/uint32" },
              "nym_id": { "string" },
              "master_id": { "string" },
              "mode": { "$ref": "#/defs/AuthorityMode" },
              "index": { "$ref": "defs.schema.json#/defs/uint32" },
              "master_credential": { "$ref": "credential.schema.json#" },
              "active_child_ids": { "$ref": "defs.schema.json#/defs/
ArrayOfStrings" },
              "active_children" : { "$ref": "#/defs/CredentialArray" },
              "revoked_child_ids": { "$ref": "defs.schema.json#/defs/
ArrayOfStrings" },
              "revoked_children" : { "$ref": "#/defs/CredentialArray" }
          },
          "required": ["version", "nym_id", "master_id", "mode", "index"],
          "additionalProperties": false
      },
      "AuthorityArray": {
          "type": "array",
          "items": {
              "credential": { "$ref": "#/defs/Authority" }
          }
      }
  }
}
```

nym_defs.json

```
{
    "version": 1,
    "nym_mode_map": {
        "NYM_ERROR" : 0,
```

```
        "NYM_PRIVATE" : 1,
        "NYM_PUBLIC" : 2
    },
    "source_type_map": {
        "SOURCETYPE_ERROR" : 0,
        "SOURCETYPE_PUBKEY" : 1,
        "SOURCETYPE_BIP47" : 2
    },
    "authority_mode_map": {
        "AUTHORITYMODE_ERROR" : 0,
        "AUTHORITYMODE_INDEX" : 1,
        "AUTHORITYMODE_FULL" : 2
    },
    "key_id": {
        "asd876asd766asd"
    },
    "signature": {
        "kjhasdf987q345kjh3q4kj5hadfs8yaskjakjhsfdi8uy34kjkjafdkjhasd"
    }
}
```

Signature

The opentxs signature schema.

signature.schema.json

```
{
  "$id": "https://matterfi.com/schemas/signature",
  "description": "Signature schema used in opentxs",
  "type": "object",
  "properties": {
    "version": { "$ref": "defs.schema.json#/defs/uint32" },
    "credential_id": { "string" },
    "role": { "$ref": "signature_defs.schema.json#/defs/SignatureRole" },
    "hash_type": { "$ref": "signature_defs.schema.json#/defs/HashType" },
    "signature": { "string" }
  },
  "required": ["version", "credential_id", "role", "hash_type", "signature"],
  "additionalProperties": false
}
```

signature_defs.schema.json

```
{
  "description": "Signature-related definitions used in opentxs",
  "type": "object",
  "properties": {
    "version": { "$ref": "defs.schema.json#/defs/uint32" },
    "signature_role_map": { "$ref": "#/$defs/mapStringSignatureRole" },
    "hash_type_map": { "$ref": "#/$defs/mapStringHashType" },
    "key_id": { "string" },
    "signature": { "string" }
  },
  "required": ["version", "signature_role_map", "hash_type_map", "key_id",
```

```
    "signature"],
  "additionalProperties": false,
  "$defs": {
      "SignatureRole": { "type": "integer", "minimum" : 0, "maximum" : 13 },
      "mapStringSignatureRole": {
          "type": "object",
          "patternProperties": {
              ".{1,}": { "$ref": "#/$defs/SignatureRole" }
          }
      },
      "HashType": { "type": "integer", "minimum" : 0, "maximum" : 12 },
      "mapStringHashType": {
          "type": "object",
          "patternProperties": {
              ".{1,}": { "$ref": "#/$defs/HashType" }
          }
      }
  }
}
```

signature_defs.json

```
{
    "version": 1,
    "signature_role_map": {
        "SIGROLE_ERROR" : 0,
        "SIGROLE_PUBCREDENTIAL" : 1,
        "SIGROLE_PRIVCREDENTIAL" : 2,
        "SIGROLE_NYMIDSOURCE" : 3,
        "SIGROLE_CLAIM" : 4,
        "SIGROLE_SERVERCONTRACT" : 5,
        "SIGROLE_UNITDEFINITION" : 6,
        "SIGROLE_PEERREQUEST" : 7,
        "SIGROLE_PEERREPLY" : 8,
        "SIGROLE_CONTEXT" : 9,
```

```
        "SIGROLE_ACCOUNT" : 10,
        "SIGROLE_SERVERREQUEST" : 11,
        "SIGROLE_SERVERREPLY" : 12,
        "SIGROLE_NYM" : 13
    },
    "hash_type_map": {
        "HASHTYPE_ERROR" : 0,
        "HASHTYPE_NONE" : 1,
        "HASHTYPE_SHA256" : 2,
        "HASHTYPE_SHA512" : 3,
        "HASHTYPE_BLAKE2B160" : 4,
        "HASHTYPE_BLAKE2B256" : 5,
        "HASHTYPE_BLAKE2B512" : 6,
        "HASHTYPE_RIPEMD160" : 7,
        "HASHTYPE_SHA1" : 8,
        "HASHTYPE_SHA256D" : 9,
        "HASHTYPE_SHA256DC" : 10,
        "HASHTYPE_BITCOIN" : 11,
        "HASHTYPE_SIPHASH24" : 12
    },
    "key_id": {
        "asd876asd766asd"
    },
    "signature": {
        "kjhasdf987q345kjh3q4kj5hadfs8yaskjakjhsfdi8uy34kjkjafdkjhasd"
    }
}
```

Key

The opentxs key schema for symmetric and asymmetric keys.

key_defs.schema.json

```
{
  "description": "Key-related definitions used in opentxs",
  "type": "object",
  "properties": {
    "version": { "$ref": "defs.schema.json#/defs/uint32" },
    "asymmetric_key_type_map": { "$ref": "#/$defs/mapStringAsymmetricKeyType" },
    "symmetric_key_type_map": { "$ref": "#/$defs/mapStringSymmetricKeyType" },
    "symmetric_mode_map": { "$ref": "#/$defs/mapStringSymmetricMode" },
    "key_mode_map": { "$ref": "#/$defs/mapStringKeyMode" },
    "key_role_map": { "$ref": "#/$defs/mapStringKeyRole" },
    "key_id": { "string" },
    "signature": { "string" }
  },
  "required": ["version", "asymmetric_key_type_map", "symmetric_key_type_map",
      "symmetric_mode_map", "key_mode_map", "key_role_map", "key_id", "signature"],
  "additionalProperties": false,
  "$defs": {
      "AsymmetricKeyType": { "type": "integer", "minimum" : 0, "maximum" : 4 },
      "mapStringAsymmetricKeyType": {
          "type": "object",
          "patternProperties": {
              ".{1,}": { "$ref": "#/$defs/AsymmetricKeyType" }
          }
      },
      "SymmetricKeyType": { "type": "integer", "minimum" : 0, "maximum" : 4 },
      "mapStringSymmetricKeyType": {
          "type": "object",
```

```
    "patternProperties": {
        ".{1,}": { "$ref": "#/$defs/SymmetricKeyType" }
    }
},
"SymmetricMode": { "type": "integer", "minimum" : 0, "maximum" : 1 },
"mapStringSymmetricMode": {
    "type": "object",
    "patternProperties": {
        ".{1,}": { "$ref": "#/$defs/SymmetricMode" }
    }
},
"KeyMode": { "type": "integer", "minimum" : 0, "maximum" : 3 },
"mapStringKeyMode": {
    "type": "object",
    "patternProperties": {
        ".{1,}": { "$ref": "#/$defs/KeyMode" }
    }
},
"KeyRole": { "type": "integer", "minimum" : 0, "maximum" : 3 },
"mapStringKeyRole": {
    "type": "object",
    "patternProperties": {
        ".{1,}": { "$ref": "#/$defs/KeyRole" }
    }
},
"HDPath": {
    "type": "object",
    "description": "todo: 'required' field to be updated",
    "properties": {
        "version": { "$ref": "defs.schema.json#/defs/uint32" },
        "root": { "string" },
        "child": {
            "type": "array",
            "items": {
```

```
                "type": { "$ref": "defs.schema.json#/defs/uint32" }
            }
        }
    },
    "required": ["version"]
},
"Ciphertext": {
    "type": "object",
    "description": "todo: 'required' field to be updated",
    "properties": {
        "version": { "$ref": "defs.schema.json#/defs/uint32" },
        "mode": { "$ref": "#/$defs/SymmetricMode" },
        "is_payload": { "boolean" },
        "key": { "$ref": "#/defs/SymmetricKey" },
        "iv": { "string" },
        "tag": { "string" },
        "data": { "string" }
    },
    "required": ["version"]
},
"SymmetricKey": {
    "type": "object",
    "description": "todo: 'required' field to be updated",
    "properties": {
        "version": { "$ref": "defs.schema.json#/defs/uint32" },
        "symmetric_key_type": { "$ref": "#/$defs/SymmetricKeyType" },
        "size": { "$ref": "defs.schema.json#/defs/uint32" },
        "salt": { "string" },
        "operations": { "$ref": "defs.schema.json#/defs/uint64" },
        "difficulty": { "$ref": "defs.schema.json#/defs/uint64" },
        "key": { "$ref": "#/defs/Ciphertext" },
        "parallel": { "$ref": "defs.schema.json#/defs/uint64" }
    },
    "required": ["version", "symmetric_key_type"]
```

```
      },
      "AsymmetricKey": {
          "type": "object",
          "description": "todo: 'required' field to be updated",
          "properties": {
              "version": { "$ref": "defs.schema.json#/defs/uint32" },
              "asymmetric_key_type": { "$ref": "#/$defs/AsymmetricKeyType" },
              "mode": { "$ref": "#/$defs/KeyMode" },
              "role": { "$ref": "#/$defs/KeyRole" },
              "key": { "string" },
              "encryptedKey": { "$ref": "#/$defs/Ciphertext" },
              "chaincode": { "$ref": "#/$defs/Ciphertext" },
              "path": { "$ref": "#/$defs/HDPath" },
              "bip32_parent": { "$ref": "defs.schema.json#/defs/uint32" },
              "params": { "string" }
          },
          "required": ["version", "asymmetric_key_type"]
      }
  }
}
```

key_defs.json

```
{
    "version": 1,
    "asymmetric_key_type_map": {
        "AKEYTYPE_ERROR" : 0,
        "AKEYTYPE_NULL" : 1,
        "AKEYTYPE_LEGACY" : 2,
        "AKEYTYPE_SECP256K1" : 3,
        "AKEYTYPE_ED25519" : 4
    },
    "symmetric_key_type_map": {
        "SKEYTYPE_ERROR" : 0,
        "SKEYTYPE_RAW" : 1,
```

```
        "SKEYTYPE_ECDH" : 2,
        "SKEYTYPE_ARGON2" : 3,
        "SKEYTYPE_ARGON2ID" : 4
    },
    "symmetric_mode_map": {
        "SMODE_ERROR" : 0,
        "SMODE_CHACHA20POLY1305" : 1
    },
    "key_mode_map": {
        "KEYMODE_ERROR" : 0,
        "KEYMODE_NULL" : 1,
        "KEYMODE_PUBLIC" : 2,
        "KEYMODE_PRIVATE" : 3
    },
    "key_role_map": {
        "KEYROLE_ERROR" : 0,
        "KEYROLE_AUTH" : 1,
        "KEYROLE_ENCRYPT" : 2,
        "KEYROLE_SIGN" : 3
    },
    "key_id": {
        "asd876asd766asd"
    },
    "signature": {
        "kjhasdf987q345kjh3q4kj5hadfs8yaskjakjhsfdi8uy34kjkjafdkjhasd"
    }
}
```

Payment Code

The opentxs payment code schema.

payment_code.schema.json

```
{
  "$id": "https://matterfi.com/schemas/payment_code",
  "description": "Payment code schema used in opentxs",
  "type": "object",
  "properties": {
    "version": { "$ref": "defs.schema.json#/defs/uint32" },
    "key": { "string" },
    "chaincode": { "string" },
    "bitmessage": { "boolean" },
    "bitmessage_version": { "$ref": "defs.schema.json#/defs/uint32" },
    "bitmessage_stream": { "$ref": "defs.schema.json#/defs/uint32" }
  },
  "required": ["version", "key", "bitmessage"],
  "additionalProperties": false
}
```

Credential

The opentxs credential schema.

credential.schema.json

```
{
  "$id": "https://matterfi.com/schemas/credential",
  "description": "Credential schema used in opentxs",
  "type": "object",
  "properties": {
    "version": { "$ref": "defs.schema.json#/defs/uint32" },
    "credential_id": { "string" },
    "credential_type": { "$ref": "credential_defs.schema.json#/defs/
CredentialType" },
    "credential_role": { "$ref": "credential_defs.schema.json#/defs/
CredentialRole" },
    "mode": { "$ref": "key_defs.schema.json#/defs/KeyMode" },
    "nym_id": { "string" },
    "child_data": { "$ref": "credential_defs.schema.json#/defs/
ChildCredentialParameters" },
    "master_data": { "$ref": "credential_defs.schema.json#/defs/
MasterCredentialParameters" },
    "public_data": {
        "type": "object",
        "properties": {
            "public_credential": { "$ref": "credential_defs.schema.json#/defs/
KeyCredential" },
            "contact_data": { "$ref": "contact_defs.schema.json#/defs/
ContactData" },
            "verification": { "$ref": "credential_defs.schema.json#/defs/
VerificationSet" }
        }
```

```
    },
    "private_data": {
        "type": "object",
        "properties": {
            "private_credential": { "$ref": "credential_defs.schema.json#/defs/
KeyCredential" }
        }
    },
    "signatures": { "$ref": "#/defs/SignatureArray" }
  },
  "required": ["version", "credential_id", "credential_type", "credential_role",
      "mode", "nym_id", "child_data", "master_data", "public_data",
      "private_data", "signatures"],
  "additionalProperties": false,
  "$defs": {
      "SignatureArray": {
          "type": "array",
          "items": { "$ref": "signature.schema.json#" }
      }
  }
}
```

credential_defs.schema.json

```
{
  "description": "Credential-related definitions used in opentxs",
  "type": "object",
  "properties": {
    "version": { "$ref": "defs.schema.json#/defs/uint32" },
    "credential_type_map": { "$ref": "#/$defs/mapStringCredentialType" },
    "credential_role_map": { "$ref": "#/$defs/mapStringCredentialRole" },
    "source_proof_type": { "$ref": "#/$defs/mapStringSourceProofType" },
    "key_id": { "string" },
    "signature": { "string" }
  },
```

```
"required": ["version", "credential_type_map", "credential_role_map",
    "key_id", "signature"],
"additionalProperties": false,
"$defs": {
    "CredentialType": { "type": "integer", "minimum" : 0, "maximum" : 2 },
    "mapStringCredentialType": {
        "type": "object",
        "patternProperties": {
            ".{1,}": { "$ref": "#/$defs/CredentialType" }
        }
    },
    "CredentialRole": { "type": "integer", "minimum" : 0, "maximum" : 4 },
    "mapStringCredentialRole": {
        "type": "object",
        "patternProperties": {
            ".{1,}": { "$ref": "#/$defs/CredentialRole" }
        }
    },
    "SourceProofType": { "type": "integer", "minimum" : 0, "maximum" : 2 },
    "mapStringSourceProofType": {
        "type": "object",
        "patternProperties": {
            ".{1,}": { "$ref": "#/$defs/SourceProofType" }
        }
    },
    "SourceProof": {
        "type": "object",
        "properties": {
            "version": { "$ref": "defs.schema.json#/defs/uint32" },
            "type": { "$ref": "#/defs/SourceProofType" }
        }
    },
    "ChildCredentialParameters": {
        "type": "object",
```

```
            "properties": {
                "version": { "$ref": "defs.schema.json#/defs/uint32" },
                "master_id": { "string" }
            },
            "required": ["version", "master_id"],
            "additionalProperties": false
        },
        "MasterCredentialParameters": {
            "type": "object",
            "properties": {
                "version": { "$ref": "defs.schema.json#/defs/uint32" },
                "source": { "$ref": "nym_defs.schema.json#/defs/NymIDSource" },
                "source_proof": { "$ref": "#/defs/SourceProof" }
            },
            "required": ["version", "source", "source_proof"],
            "additionalProperties": false
        },
        "KeyCredential": {
            "type": "object",
            "properties": {
                "version": { "$ref": "defs.schema.json#/defs/uint32" },
                "mode": { "$ref": "key_defs.schema.json#/defs/KeyMode" },
                "key": { "$ref": "key_defs.schema.json#/defs/AsymmetricKey" }
            },
            "required": ["version", "mode", "key"],
            "additionalProperties": false
        }
    }
}
```

credential_defs.json

```
{
    "version": 1,
    "credential_type_map": {
```

```
        "CREDTYPE_ERROR" : 0,
        "CREDTYPE_LEGACY" : 1,
        "CREDTYPE_HD" : 2
    },
    "credential_role_map": {
        "CREDROLE_ERROR" : 0,
        "CREDROLE_MASTERKEY" : 1,
        "CREDROLE_CHILDKEY" : 2,
        "CREDROLE_CONTACT" : 3,
        "CREDROLE_VERIFY" : 4
    },
    "source_proof_type": {
        "SOURCEPROOFTYPE_ERROR" : 0,
        "SOURCEPROOFTYPE_SELF_SIGNATURE" : 1,
        "SOURCEPROOFTYPE_SIGNATURE" : 2
    },
    "key_id": {
        "asd876asd766asd"
    },
    "signature": {
        "kjhasdf987q345kjh3q4kj5hadfs8yaskjakjhsfdi8uy34kjkjafdkjhasd"
    }
}
```

Contact

The opentxs contact schema.

contact_defs.schema.json

```
{
  "description": "Contact Item (credential)-related definitions used in opentxs",
  "type": "object",
  "properties": {
    "version": { "$ref": "defs.schema.json#/defs/uint32" },
    "contact_section_name_map": { "$ref": "#/$defs/mapStringContactSectionName" },
    "contact_item_type_map": { "$ref": "#/$defs/mapStringContactItemType" },
    "contact_item_attribute_map": { "$ref": "#/$defs/
mapStringContactItemAttribute" },
    "key_id": { "string" },
    "signature": { "string" }
  },
  "required": ["version", "contact_section_name_map", "contact_item_type_map",
      "contact_item_attribute_map", "key_id", "signature"],
  "additionalProperties": false,
  "$defs": {
      "ContactSectionName": { "type": "integer", "minimum" : 0, "maximum" : 13 },
      "mapStringContactSectionName": {
          "type": "object",
          "patternProperties": {
              ".{1,}": { "$ref": "#/$defs/ContactSectionName" }
          }
      },
      "ContactItemType": { "type": "integer", "minimum" : 0, "maximum" : 425 },
      "mapStringContactItemType": {
          "type": "object",
          "patternProperties": {
```

```
        ".{1,}": { "$ref": "#/$defs/ContactItemType" }
    }
},
"ContactItemAttribute": { "type": "integer", "minimum" : 0, "maximum" : 3 },
"mapStringContactItemAttribute": {
    "type": "object",
    "patternProperties": {
        ".{1,}": { "$ref": "#/$defs/ContactItemAttribute" }
    }
},
"ContactData": {
    "type": "object",
    "properties": {
        "version": { "$ref": "defs.schema.json#/defs/uint32" },
        "section": { "$ref": "#/defs/ContactSection" }
    },
    "required": ["version", "section"],
    "additionalProperties": false
},
"ContactSection": {
    "type": "object",
    "properties": {
        "version": { "$ref": "defs.schema.json#/defs/uint32" },
        "section_name": { "$ref": "#/defs/ContactSectionName" },
        "section_item": { "$ref": "#/defs/ContactItem" }
    },
    "required": ["version", "section_name", "section_item"],
    "additionalProperties": false
},
"ContactItem": {
    "type": "object",
    "properties": {
        "version": { "$ref": "defs.schema.json#/defs/uint32" },
        "item_id": { "string" },
```

```
        "item_type": { "$ref": "#/defs/ContactItemType" },
        "item_value": { "string" },
        "start": { "$ref": "defs.schema.json#/defs/int64" },
        "end": { "$ref": "defs.schema.json#/defs/int64" },
        "attributes": {
            "type": "array",
            "items": { "$ref": "#/$defs/ContactItemAttribute" }
        },
        "subtype": { "string" }
    },
    "required": ["version", "item_id", "item_type", "item_value",
        "start", "end", "attributes", "subtype"],
    "additionalProperties": false
},
"Verification": {
    "type": "object",
    "properties": {
        "version": { "$ref": "defs.schema.json#/defs/uint32" },
        "verification_id": { "string" },
        "claim": { "string" },
        "valid": { "boolean" },
        "start": { "$ref": "defs.schema.json#/defs/int64" },
        "end": { "$ref": "defs.schema.json#/defs/int64" },
        "signature": { "$ref": "signature.schema.json#" },
        "retracted": { "boolean" }
    },
    "required": ["version", "verification_id", "claim", "valid",
        "start", "end", "signature", "retracted"],
    "additionalProperties": false
},
"VerificationIdentity": {
    "type": "object",
    "properties": {
        "version": { "$ref": "defs.schema.json#/defs/uint32" },
```

```
            "nym": { "string" },
            "verification": { "$ref": "#/defs/Verification" }
        },
        "required": ["version", "nym", "verification"],
        "additionalProperties": false
    },
    "VerificationGroup": {
        "type": "object",
        "properties": {
            "version": { "$ref": "defs.schema.json#/defs/uint32" },
            "identity": { "$ref": "#/defs/VerificationIdentity" }
        },
        "required": ["version", "identity"],
        "additionalProperties": false
    },
    "VerificationSet": {
        "type": "object",
        "properties": {
            "version": { "$ref": "defs.schema.json#/defs/uint32" },
            "internal": { "$ref": "#/defs/VerificationGroup" },
            "external": { "$ref": "#/defs/VerificationGroup" },
            "repudiated": {
                "type": "array",
                "items": { "string" }
            },
        },
        "required": ["version", "internal", "external", "repudiated"],
        "additionalProperties": false
    }
  }
}
```

contact_defs.json

```
{
```

```
"version": 1,
"contact_section_name_map": {
    "CONTACTSECTION_ERROR" : 0,
    "CONTACTSECTION_SCOPE" : 1,
    "CONTACTSECTION_IDENTIFIER" : 2,
    "CONTACTSECTION_ADDRESS" : 3,
    "CONTACTSECTION_COMMUNICATION" : 4,
    "CONTACTSECTION_PROFILE" : 5,
    "CONTACTSECTION_RELATIONSHIP" : 6,
    "CONTACTSECTION_DESCRIPTOR" : 7,
    "CONTACTSECTION_EVENT" : 8,
    "CONTACTSECTION_CONTRACT" : 9,
    "CONTACTSECTION_PROCEDURE" : 10
},
"contact_item_type_map": {
    "CITEMTYPE_ERROR" : 0,
    "CITEMTYPE_INDIVIDUAL" : 1,
    "CITEMTYPE_ORGANIZATION" : 2,
    "CITEMTYPE_BUSINESS" : 3,
    "CITEMTYPE_GOVERNMENT" : 4,
    "CITEMTYPE_SERVER" : 5,
    "CITEMTYPE_PREFIX" : 6,
    "CITEMTYPE_FORENAME" : 7,
    "CITEMTYPE_MIDDLENAME" : 8,
    "CITEMTYPE_SURNAME" : 9,
    "CITEMTYPE_PEDIGREE" : 10,
    "CITEMTYPE_SUFFIX" : 11,
    "CITEMTYPE_NICKNAME" : 12,
    "CITEMTYPE_COMMONNAME" : 13,
    "CITEMTYPE_PASSPORT" : 14,
    "CITEMTYPE_NATIONAL" : 15,
    "CITEMTYPE_PROVINCIAL" : 16,
    "CITEMTYPE_MILITARY" : 17,
    "CITEMTYPE_PGP" : 18,
```

```
"CITEMTYPE_OTR" : 19,
"CITEMTYPE_SSL" : 20,
"CITEMTYPE_PHYSICAL" : 21,
"CITEMTYPE_OFFICIAL" : 22,
"CITEMTYPE_BIRTHPLACE" : 23,
"CITEMTYPE_HOME" : 24,
"CITEMTYPE_WEBSITE" : 25,
"CITEMTYPE_OPENTXS" : 26,
"CITEMTYPE_PHONE" : 27,
"CITEMTYPE_EMAIL" : 28,
"CITEMTYPE_SKYPE" : 29,
"CITEMTYPE_WIRE" : 30,
"CITEMTYPE_QQ" : 31,
"CITEMTYPE_BITMESSAGE" : 32,
"CITEMTYPE_WHATSAPP" : 33,
"CITEMTYPE_TELEGRAM" : 34,
"CITEMTYPE_KIK" : 35,
"CITEMTYPE_BBM" : 36,
"CITEMTYPE_WECHAT" : 37,
"CITEMTYPE_KAKAOTALK" : 38,
"CITEMTYPE_FACEBOOK" : 39,
"CITEMTYPE_GOOGLE" : 40,
"CITEMTYPE_LINKEDIN" : 41,
"CITEMTYPE_VK" : 42,
"CITEMTYPE_ABOUTME" : 43,
"CITEMTYPE_ONENAME" : 44,
"CITEMTYPE_TWITTER" : 45,
"CITEMTYPE_MEDIUM" : 46,
"CITEMTYPE_TUMBLR" : 47,
"CITEMTYPE_YAHOO" : 48,
"CITEMTYPE_MYSPACE" : 49,
"CITEMTYPE_MEETUP" : 50,
"CITEMTYPE_REDDIT" : 51,
"CITEMTYPE_HACKERNEWS" : 52,
```

```
"CITEMTYPE_WIKIPEDIA" : 53,
"CITEMTYPE_ANGELLIST" : 54,
"CITEMTYPE_GITHUB" : 55,
"CITEMTYPE_BITBUCKET" : 56,
"CITEMTYPE_YOUTUBE" : 57,
"CITEMTYPE_VIMEO" : 58,
"CITEMTYPE_TWITCH" : 59,
"CITEMTYPE_SNAPCHAT" : 60,
"CITEMTYPE_VINE" : 61,
"CITEMTYPE_INSTAGRAM" : 62,
"CITEMTYPE_PINTEREST" : 63,
"CITEMTYPE_IMGUR" : 64,
"CITEMTYPE_FLICKR" : 65,
"CITEMTYPE_DRIBBLE" : 66,
"CITEMTYPE_BEHANCE" : 67,
"CITEMTYPE_DEVIANTART" : 68,
"CITEMTYPE_SPOTIFY" : 69,
"CITEMTYPE_ITUNES" : 70,
"CITEMTYPE_SOUNDCLOUD" : 71,
"CITEMTYPE_ASKFM" : 72,
"CITEMTYPE_EBAY" : 73,
"CITEMTYPE_ETSY" : 74,
"CITEMTYPE_OPENBAZAAR" : 75,
"CITEMTYPE_XBOXLIVE" : 76,
"CITEMTYPE_PLAYSTATION" : 77,
"CITEMTYPE_SECONDLIFE" : 78,
"CITEMTYPE_WARCRAFT" : 79,
"CITEMTYPE_ALIAS" : 80,
"CITEMTYPE_ACQUAINTANCE" : 81,
"CITEMTYPE_FRIEND" : 82,
"CITEMTYPE_SPOUSE" : 83,
"CITEMTYPE_SIBLING" : 84,
"CITEMTYPE_MEMBER" : 85,
"CITEMTYPE_COLLEAGUE" : 86,
```

```
"CITEMTYPE_PARENT" : 87,
"CITEMTYPE_CHILD" : 88,
"CITEMTYPE_EMPLOYER" : 89,
"CITEMTYPE_EMPLOYEE" : 90,
"CITEMTYPE_CITIZEN" : 91,
"CITEMTYPE_PHOTO" : 92,
"CITEMTYPE_GENDER" : 93,
"CITEMTYPE_HEIGHT" : 94,
"CITEMTYPE_WEIGHT" : 95,
"CITEMTYPE_HAIR" : 96,
"CITEMTYPE_EYE" : 97,
"CITEMTYPE_SKIN" : 98,
"CITEMTYPE_ETHNICITY" : 99,
"CITEMTYPE_LANGUAGE" : 100,
"CITEMTYPE_DEGREE" : 101,
"CITEMTYPE_CERTIFICATION" : 102,
"CITEMTYPE_TITLE" : 103,
"CITEMTYPE_SKILL" : 104,
"CITEMTYPE_AWARD" : 105,
"CITEMTYPE_LIKES" : 106,
"CITEMTYPE_SEXUAL" : 107,
"CITEMTYPE_POLITICAL" : 108,
"CITEMTYPE_RELIGIOUS" : 109,
"CITEMTYPE_BIRTH" : 110,
"CITEMTYPE_SECONDARYGRADUATION" : 111,
"CITEMTYPE_UNIVERSITYGRADUATION" : 112,
"CITEMTYPE_WEDDING" : 113,
"CITEMTYPE_ACCOMPLISHMENT" : 114,
"CITEMTYPE_BTC" : 115,
"CITEMTYPE_ETH" : 116,
"CITEMTYPE_XRP" : 117,
"CITEMTYPE_LTC" : 118,
"CITEMTYPE_DAO" : 119,
"CITEMTYPE_XEM" : 120,
```

```
"CITEMTYPE_DASH" : 121,
"CITEMTYPE_MAID" : 122,
"CITEMTYPE_LSK" : 123,
"CITEMTYPE_DOGE" : 124,
"CITEMTYPE_DGD" : 125,
"CITEMTYPE_XMR" : 126,
"CITEMTYPE_WAVES" : 127,
"CITEMTYPE_NXT" : 128,
"CITEMTYPE_SC" : 129,
"CITEMTYPE_STEEM" : 130,
"CITEMTYPE_AMP" : 131,
"CITEMTYPE_XLM" : 132,
"CITEMTYPE_FCT" : 133,
"CITEMTYPE_BTS" : 134,
"CITEMTYPE_USD" : 135,
"CITEMTYPE_EUR" : 136,
"CITEMTYPE_GBP" : 137,
"CITEMTYPE_INR" : 138,
"CITEMTYPE_AUD" : 139,
"CITEMTYPE_CAD" : 140,
"CITEMTYPE_SGD" : 141,
"CITEMTYPE_CHF" : 142,
"CITEMTYPE_MYR" : 143,
"CITEMTYPE_JPY" : 144,
"CITEMTYPE_CNY" : 145,
"CITEMTYPE_NZD" : 146,
"CITEMTYPE_THB" : 147,
"CITEMTYPE_HUF" : 148,
"CITEMTYPE_AED" : 149,
"CITEMTYPE_HKD" : 150,
"CITEMTYPE_MXN" : 151,
"CITEMTYPE_ZAR" : 152,
"CITEMTYPE_PHP" : 153,
"CITEMTYPE_SEK" : 154,
```

```
"CITEMTYPE_TNBTC" : 155,
"CITEMTYPE_TNXRP" : 156,
"CITEMTYPE_TNLTC" : 157,
"CITEMTYPE_TNXEM" : 158,
"CITEMTYPE_TNDASH" : 159,
"CITEMTYPE_TNMAID" : 160,
"CITEMTYPE_TNLSK" : 161,
"CITEMTYPE_TNDOGE" : 162,
"CITEMTYPE_TNXMR" : 163,
"CITEMTYPE_TNWAVES" : 164,
"CITEMTYPE_TNNXT" : 165,
"CITEMTYPE_TNSC" : 166,
"CITEMTYPE_TNSTEEM" : 167,
"CITEMTYPE_PHILOSOPHY" : 168,
"CITEMTYPE_MET" : 169,
"CITEMTYPE_FAN" : 170,
"CITEMTYPE_SUPERVISOR" : 171,
"CITEMTYPE_SUBORDINATE" : 172,
"CITEMTYPE_CONTACT" : 173,
"CITEMTYPE_REFRESHED" : 174,
"CITEMTYPE_BOT" : 175,
"CITEMTYPE_BCH" : 176,
"CITEMTYPE_TNBCH" : 177,
"CITEMTYPE_OWNER" : 178,
"CITEMTYPE_PROPERTY" : 179,
"CITEMTYPE_UNKNOWN" : 180,
"CITEMTYPE_ETHEREUM_OLYMPIC" : 181,
"CITEMTYPE_ETHEREUM_CLASSIC" : 182,
"CITEMTYPE_ETHEREUM_EXPANSE" : 183,
"CITEMTYPE_ETHEREUM_MORDEN" : 184,
"CITEMTYPE_ETHEREUM_ROPSTEN" : 185,
"CITEMTYPE_ETHEREUM_RINKEBY" : 186,
"CITEMTYPE_ETHEREUM_KOVAN" : 187,
"CITEMTYPE_ETHEREUM_SOKOL" : 188,
```

```
"CITEMTYPE_ETHEREUM_POA" : 189,
"CITEMTYPE_PKT" : 190,
"CITEMTYPE_TNPKT" : 191,
"CITEMTYPE_REGTEST" : 192,
"CITEMTYPE_BNB" : 193,
"CITEMTYPE_SOL" : 194,
"CITEMTYPE_USDT" : 195,
"CITEMTYPE_ADA" : 196,
"CITEMTYPE_DOT" : 197,
"CITEMTYPE_USDC" : 198,
"CITEMTYPE_SHIB" : 199,
"CITEMTYPE_LUNA" : 200,
"CITEMTYPE_AVAX" : 201,
"CITEMTYPE_UNI" : 202,
"CITEMTYPE_LINK" : 203,
"CITEMTYPE_WBTC" : 204,
"CITEMTYPE_BUSD" : 205,
"CITEMTYPE_MATIC" : 206,
"CITEMTYPE_ALGO" : 207,
"CITEMTYPE_VET" : 208,
"CITEMTYPE_AXS" : 209,
"CITEMTYPE_ICP" : 210,
"CITEMTYPE_CRO" : 211,
"CITEMTYPE_ATOM" : 212,
"CITEMTYPE_THETA" : 213,
"CITEMTYPE_FIL" : 214,
"CITEMTYPE_TRX" : 215,
"CITEMTYPE_FTT" : 216,
"CITEMTYPE_ETC" : 217,
"CITEMTYPE_FTM" : 218,
"CITEMTYPE_DAI" : 219,
"CITEMTYPE_BTCB" : 220,
"CITEMTYPE_EGLD" : 221,
"CITEMTYPE_HBAR" : 222,
```

```
"CITEMTYPE_XTZ" : 223,
"CITEMTYPE_MANA" : 224,
"CITEMTYPE_NEAR" : 225,
"CITEMTYPE_GRT" : 226,
"CITEMTYPE_CAKE" : 227,
"CITEMTYPE_EOS" : 228,
"CITEMTYPE_FLOW" : 229,
"CITEMTYPE_AAVE" : 230,
"CITEMTYPE_KLAY" : 231,
"CITEMTYPE_KSM" : 232,
"CITEMTYPE_XEC" : 233,
"CITEMTYPE_MIOTA" : 234,
"CITEMTYPE_HNT" : 235,
"CITEMTYPE_RUNE" : 236,
"CITEMTYPE_BSV" : 237,
"CITEMTYPE_LEO" : 238,
"CITEMTYPE_NEO" : 239,
"CITEMTYPE_ONE" : 240,
"CITEMTYPE_QNT" : 241,
"CITEMTYPE_UST" : 242,
"CITEMTYPE_MKR" : 243,
"CITEMTYPE_ENJ" : 244,
"CITEMTYPE_CHZ" : 245,
"CITEMTYPE_AR" : 246,
"CITEMTYPE_STX" : 247,
"CITEMTYPE_BTT" : 248,
"CITEMTYPE_HOT" : 249,
"CITEMTYPE_SAND" : 250,
"CITEMTYPE_OMG" : 251,
"CITEMTYPE_CELO" : 252,
"CITEMTYPE_ZEC" : 253,
"CITEMTYPE_COMP" : 254,
"CITEMTYPE_TFUEL" : 255,
"CITEMTYPE_KDA" : 256,
```

```
"CITEMTYPE_LRC" : 257,
"CITEMTYPE_QTUM" : 258,
"CITEMTYPE_CRV" : 259,
"CITEMTYPE_HT" : 260,
"CITEMTYPE_NEXO" : 261,
"CITEMTYPE_SUSHI" : 262,
"CITEMTYPE_KCS" : 263,
"CITEMTYPE_BAT" : 264,
"CITEMTYPE_OKB" : 265,
"CITEMTYPE_DCR" : 266,
"CITEMTYPE_ICX" : 267,
"CITEMTYPE_RVN" : 268,
"CITEMTYPE_SCRT" : 269,
"CITEMTYPE_REV" : 270,
"CITEMTYPE_AUDIO" : 271,
"CITEMTYPE_ZIL" : 272,
"CITEMTYPE_TUSD" : 273,
"CITEMTYPE_YFI" : 274,
"CITEMTYPE_MINA" : 275,
"CITEMTYPE_PERP" : 276,
"CITEMTYPE_XDC" : 277,
"CITEMTYPE_TEL" : 278,
"CITEMTYPE_SNX" : 279,
"CITEMTYPE_BTG" : 280,
"CITEMTYPE_AFN" : 281,
"CITEMTYPE_ALL" : 282,
"CITEMTYPE_AMD" : 283,
"CITEMTYPE_ANG" : 284,
"CITEMTYPE_AOA" : 285,
"CITEMTYPE_ARS" : 286,
"CITEMTYPE_AWG" : 287,
"CITEMTYPE_AZN" : 288,
"CITEMTYPE_BAM" : 289,
"CITEMTYPE_BBD" : 290,
```

```
"CITEMTYPE_BDT" : 291,
"CITEMTYPE_BGN" : 292,
"CITEMTYPE_BHD" : 293,
"CITEMTYPE_BIF" : 294,
"CITEMTYPE_BMD" : 295,
"CITEMTYPE_BND" : 296,
"CITEMTYPE_BOB" : 297,
"CITEMTYPE_BRL" : 298,
"CITEMTYPE_BSD" : 299,
"CITEMTYPE_BTN" : 300,
"CITEMTYPE_BWP" : 301,
"CITEMTYPE_BYN" : 302,
"CITEMTYPE_BZD" : 303,
"CITEMTYPE_CDF" : 304,
"CITEMTYPE_CLP" : 305,
"CITEMTYPE_COP" : 306,
"CITEMTYPE_CRC" : 307,
"CITEMTYPE_CUC" : 308,
"CITEMTYPE_CUP" : 309,
"CITEMTYPE_CVE" : 310,
"CITEMTYPE_CZK" : 311,
"CITEMTYPE_DJF" : 312,
"CITEMTYPE_DKK" : 313,
"CITEMTYPE_DOP" : 314,
"CITEMTYPE_DZD" : 315,
"CITEMTYPE_EGP" : 316,
"CITEMTYPE_ERN" : 317,
"CITEMTYPE_ETB" : 318,
"CITEMTYPE_FJD" : 319,
"CITEMTYPE_FKP" : 320,
"CITEMTYPE_GEL" : 321,
"CITEMTYPE_GGP" : 322,
"CITEMTYPE_GHS" : 323,
"CITEMTYPE_GIP" : 324,
```

```
"CITEMTYPE_GMD" : 325,
"CITEMTYPE_GNF" : 326,
"CITEMTYPE_GTQ" : 327,
"CITEMTYPE_GYD" : 328,
"CITEMTYPE_HNL" : 329,
"CITEMTYPE_HRK" : 330,
"CITEMTYPE_HTG" : 331,
"CITEMTYPE_IDR" : 332,
"CITEMTYPE_ILS" : 333,
"CITEMTYPE_IMP" : 334,
"CITEMTYPE_IQD" : 335,
"CITEMTYPE_IRR" : 336,
"CITEMTYPE_ISK" : 337,
"CITEMTYPE_JEP" : 338,
"CITEMTYPE_JMD" : 339,
"CITEMTYPE_JOD" : 340,
"CITEMTYPE_KES" : 341,
"CITEMTYPE_KGS" : 342,
"CITEMTYPE_KHR" : 343,
"CITEMTYPE_KMF" : 344,
"CITEMTYPE_KPW" : 345,
"CITEMTYPE_KRW" : 346,
"CITEMTYPE_KWD" : 347,
"CITEMTYPE_KYD" : 348,
"CITEMTYPE_KZT" : 349,
"CITEMTYPE_LAK" : 350,
"CITEMTYPE_LBP" : 351,
"CITEMTYPE_LKR" : 352,
"CITEMTYPE_LRD" : 353,
"CITEMTYPE_LSL" : 354,
"CITEMTYPE_LYD" : 355,
"CITEMTYPE_MAD" : 356,
"CITEMTYPE_MDL" : 357,
"CITEMTYPE_MGA" : 358,
```

```
"CITEMTYPE_MKD" : 359,
"CITEMTYPE_MMK" : 360,
"CITEMTYPE_MNT" : 361,
"CITEMTYPE_MOP" : 362,
"CITEMTYPE_MRU" : 363,
"CITEMTYPE_MUR" : 364,
"CITEMTYPE_MVR" : 365,
"CITEMTYPE_MWK" : 366,
"CITEMTYPE_MZN" : 367,
"CITEMTYPE_NAD" : 368,
"CITEMTYPE_NGN" : 369,
"CITEMTYPE_NIO" : 370,
"CITEMTYPE_NOK" : 371,
"CITEMTYPE_NPR" : 372,
"CITEMTYPE_OMR" : 373,
"CITEMTYPE_PAB" : 374,
"CITEMTYPE_PEN" : 375,
"CITEMTYPE_PGK" : 376,
"CITEMTYPE_PKR" : 377,
"CITEMTYPE_PLN" : 378,
"CITEMTYPE_PYG" : 379,
"CITEMTYPE_QAR" : 380,
"CITEMTYPE_RON" : 381,
"CITEMTYPE_RSD" : 382,
"CITEMTYPE_RUB" : 383,
"CITEMTYPE_RWF" : 384,
"CITEMTYPE_SAR" : 385,
"CITEMTYPE_SBD" : 386,
"CITEMTYPE_SCR" : 387,
"CITEMTYPE_SDG" : 388,
"CITEMTYPE_SHP" : 389,
"CITEMTYPE_SLL" : 390,
"CITEMTYPE_SOS" : 391,
"CITEMTYPE_SPL" : 392,
```

```
    "CITEMTYPE_SRD" : 393,
    "CITEMTYPE_STN" : 394,
    "CITEMTYPE_SVC" : 395,
    "CITEMTYPE_SYP" : 396,
    "CITEMTYPE_SZL" : 397,
    "CITEMTYPE_TJS" : 398,
    "CITEMTYPE_TMT" : 399,
    "CITEMTYPE_TND" : 400,
    "CITEMTYPE_TOP" : 401,
    "CITEMTYPE_TRY" : 402,
    "CITEMTYPE_TTD" : 403,
    "CITEMTYPE_TVD" : 404,
    "CITEMTYPE_TWD" : 405,
    "CITEMTYPE_TZS" : 406,
    "CITEMTYPE_UAH" : 407,
    "CITEMTYPE_UGX" : 408,
    "CITEMTYPE_UYU" : 409,
    "CITEMTYPE_UZS" : 410,
    "CITEMTYPE_VEF" : 411,
    "CITEMTYPE_VND" : 412,
    "CITEMTYPE_VUV" : 413,
    "CITEMTYPE_WST" : 414,
    "CITEMTYPE_XAF" : 415,
    "CITEMTYPE_XCD" : 416,
    "CITEMTYPE_XDR" : 417,
    "CITEMTYPE_XOF" : 418,
    "CITEMTYPE_XPF" : 419,
    "CITEMTYPE_YER" : 420,
    "CITEMTYPE_ZMW" : 421,
    "CITEMTYPE_ZWD" : 422,
    "CITEMTYPE_CUSTOM" : 423,
    "CITEMTYPE_TNBSV" : 424,
    "CITEMTYPE_TNXEC" : 425
},
```

```
    "contact_item_attribute_map": {
        "CITEMATTR_ERROR" : 0,
        "CITEMATTR_ACTIVE" : 1,
        "CITEMATTR_PRIMARY" : 2,
        "CITEMATTR_LOCAL" : 3
    },
    "key_id": {
        "asd876asd766asd"
    },
    "signature": {
        "kjhasdf987q345kjh3q4kj5hadfs8yaskjakjhsfdi8uy34kjkjafdkjhasd"
    }
}
```