lib Violas Client

1.0

Generated by Doxygen 1.8.17

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 2

File Index

2.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 3

Class Documentation

3.1 violas::Account Struct Reference

Public Attributes

- Address **address**
- AuthenticationKey **auth_key**
- PublicKey **pub_key**
- uint64_t **sequence_number**
- AccountStatus **status**

The documentation for this struct was generated from the following file:

- src/ffi/client.hpp

3.2 violas::AddressAndIndex Struct Reference

Public Attributes

- Address **address**
- size_t **index**

The documentation for this struct was generated from the following file:

- src/ffi/client.hpp

3.3 violas::Client Class Reference

Public Types

- enum event_type { **sent**, **received** }

  evnet type
- using **TransactionAugment** = std::variant< uint8_t, uint64_t, __uint128_t, Address, std::vector< uint8_t >, bool >

Public Member Functions

- virtual void **test_connection** ()=0
- virtual AddressAndIndex **create_next_account** (const std::optional< Address > &address=std::nullopt)=0
- virtual std::vector< Account > **get_all_accounts** ()=0
- virtual void **create_validator_account** (std::string_view currency_code, const AuthenticationKey &auth_key, std::string_view human_name)=0
- virtual void **mint_for_testnet** (std::string_view currency_code, const Address &receiver, uint64_t amount)=0
- virtual void transfer (size_t sender_account_ref_id, const Address &receiver_address, std::string_view currency_code, uint64_t amount, uint64_t gas_unit_price=0, uint64_t max_gas_amount=1000000, std::string_view gas_currency_code="LBR")=0

  transfer currency
- virtual void **allow_custom_script** ()=0
- virtual void **allow_publishing_module** (bool enabled)=0
- virtual void **publish_module** (size_t account_index, std::string_view module_file_name)=0
- virtual void execute_script (size_t account_index, std::string_view script_file_name, const std::vector< TypeTag > &type_tags={}, const std::vector< TransactionAugment > &arguments={})=0

  Execute script file with arguments.
- virtual std::string query_account_info (const Address &address)=0

  Query accout status infomation.
- virtual std::string query_transaction_info (const Address &address, uint64_t seq_number, bool is_fetching_event)=0

  Query transaction inforamtion by address and seqence number.
- virtual std::string query_transaction_info (uint64_t start_version, uint64_t limit, bool is_fetching_event)=0

  Query transaction inforamtion by range.
- virtual std::string query_events (const Address &address, event_type type, uint64_t start_version, uint64_t limit)=0

  Query events.
- virtual void **publish_curency** (std::string_view currency_code)=0
- virtual void **register_currency** (std::string_view currency_code, uint64_t exchange_rate_denom, uint64_t exchange_rate_num, bool is_synthetic, uint64_t scaling_factor, uint64_t fractional_part)=0
- virtual void add_currency_for_designated_dealer (std::string_view currency_code, const Address &dd_address)=0

  add currency for the designated dealer account
- virtual void add_currency (size_t account_index, std::string_view currency_code)=0

  Add a currency to current account.
- virtual uint64_t get_currency_balance (const Address &address, std::string_view currency_code)=0

  get the balance of currency for the account address
- virtual std::string get_all_currency_info ()=0

  Get all currency info.
- virtual void mint_currency_for_designated_dealer (std::string_view currency_code, uint64_t sliding_nonce, const Address &dd_address, uint64_t amount, uint64_t tier_index)=0

  mint currency for dd account
- virtual void **create_designated_dealer_account** (std::string_view currency_code, uint64_t nonce, const Address &new_account_address, const AuthenticationKey &auth_key, std::string_view human_name, std::string_view base_url, PublicKey compliance_public_key, bool add_all_currencies)=0
- virtual void **update_account_authentication_key** (const Address &address, const AuthenticationKey &auth_key)=0
- virtual void **create_parent_vasp_account** (std::string_view currency_code, uint64_t nonce, const Address &new_account_address, const AuthenticationKey &auth_key, std::string_view human_name, std::string_view base_url, PublicKey compliance_public_key, bool add_all_currencies)=0
- virtual std::string **get_exchange_currencies** (const Address &address)=0
- virtual std::string **get_exchange_reserves** (const Address &address)=0
- virtual std::string **get_liquidity_balance** (const Address &address)=0

Static Public Member Functions

- static std::shared_ptr< Client > **create** (uint8_t chain_id, std::string_view url, std::string_view mint_key, std::string_view mnemonic, std::string_view waypoint)

3.3.1 Member Enumeration Documentation

3.3.1.1 event_type

```
enum violas::Client::event_type
```

evnet type

3.3.2 Member Function Documentation

3.3.2.1 execute_script()

```
virtual void violas::Client::execute_script (
             size_t account_index,
             std::string_view script_file_name,
             const std::vector< TypeTag > & type_tags = {},
             const std::vector< TransactionAugment > & arguments = {} )  [pure virtual]
```

Execute script file with arguments.

Parameters

account_index	account index of wallet
script_file_name	script file name with path
type_tags	transaction TypeTag vector for script
arguments	transaction argument vector for script

3.3.2.2 query_account_info()

```
virtual std::string violas::Client::query_account_info (
             const Address & address )  [pure virtual]
```

Query accout status infomation.

Parameters

address	- the address of account

Returns

std::string

3.3.2.3 query_events()

```
virtual std::string violas::Client::query_events (
            const Address & address,
            event_type type,
            uint64_t start_version,
            uint64_t limit )  [pure virtual]
```

Query events.

Parameters

address	the address of account
type	evnet type
start_version	start version
limit	limit of rang, amount of queried events

Returns

std::string with json format

3.3.2.4 query_transaction_info() [1/2]

```
virtual std::string violas::Client::query_transaction_info (
            const Address & address,
            uint64_t seq_number,
            bool is_fetching_event )  [pure virtual]
```

Query transaction inforamtion by address and seqence number.

Parameters

address	the address of account
seq_number	the sequence number of account
is_fetching_event	whether fectching event or not

Returns

std::string with json format

3.3.2.5 query_transaction_info() [2/2]

```
virtual std::string violas::Client::query_transaction_info (
            uint64_t start_version,
            uint64_t limit,
            bool is_fetching_event )  [pure virtual]
```

Query transaction inforamtion by range.

Parameters

start_version	start version
limit	limit of range, amount of queried transaction
is_fetching_event	whether fectching event or not

Returns

std::string with json format

3.3.2.6 transfer()

```
virtual void violas::Client::transfer (
            size_t sender_account_ref_id,
            const Address & receiver_address,
            std::string_view currency_code,
            uint64_t amount,
            uint64_t gas_unit_price = 0,
            uint64_t max_gas_amount = 1000000,
            std::string_view gas_currency_code = "LBR" )  [pure virtual]
```

transfer currency

Parameters

sender_account_ref↵ _id	the account index of client's wallet
receiver_address	the address of receiver
currency_code	currency code
amount	the amount of currency
gas_unit_price	the gas unit price
max_gas_amount	the max gas amount, default is 1,000,000
gas_currency_code	the gas currency code. default is 'LBR'

The documentation for this class was generated from the following file:

- src/ffi/client.hpp

3.4 violas::TypeTag Struct Reference

Public Member Functions

- **TypeTag** (Address addr, std::string_view mod, std::string_view res)
- **TypeTag** (const TypeTag &tag)
- **TypeTag** (TypeTag &&tag)
- TypeTag & **operator=** (TypeTag &&tag)

Public Attributes

- Address **address**
- std::string **module_name**
- std::string **resource_name**

The documentation for this struct was generated from the following file:

- src/ffi/client.hpp

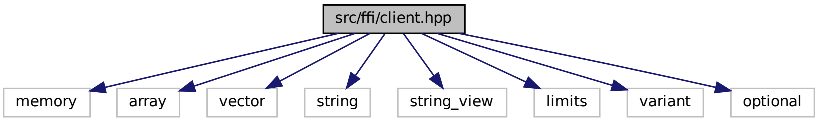
Chapter 4

File Documentation

4.1 src/ffi/client.hpp File Reference

```
#include <memory>
#include <array>
#include <vector>
#include <string>
#include <string_view>
#include <limits>
#include <variant>
#include <optional>
```

Include dependency graph for client.hpp:



Classes

- struct violas::AddressAndIndex
- struct violas::Account
- struct violas::TypeTag
- class violas::Client

Typedefs

- using **violas::Address** = std::array< uint8_t, ADDRESS_LENGTH >
- using **violas::AuthenticationKey** = std::array< uint8_t, ADDRESS_LENGTH *2 >
- using **violas::PublicKey** = std::array< uint8_t, ADDRESS_LENGTH *2 >
- using **violas::VecU8** = std::vector< uint8_t >
- using **violas::client_ptr** = std::shared_ptr< Client >

Enumerations

- enum **AccountStatus** { **Local**, **Persisted**, **Unknow** }

Functions

- TypeTag **violas::make_currency_tag** (std::string_view currency_code)

Variables

- const size_t **violas::ADDRESS_LENGTH** = 16
- const uint64_t **violas::MICRO_COIN** = 1E+6
- const uint64_t **violas::ASSOCIATION_ID** = std::numeric_limits<uint64_t>::max()
- const Address **violas::ASSOCIATION_ADDRESS** = {00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 0x0A, 0x55, 0x0C, 0x18}
- const Address **violas::TESTNET_DD_ADDRESS** = {00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 0xDD}
- const Address **violas::CORE_CODE_ADDRESS** = {00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 0x01}

4.1.1 Detailed Description

Author

Hunter Sun (HunterSun2018@gmail.com)

Version

0.1

Date

2020-09-17

Copyright

Index