PEPITAS CRYPTOCURRENCY

Generated by Doxygen 1.8.17

1 PEPITAS 1

1.1 CODING STYLE . . . 1

1.1.1 Coding case . . . 1

1.1.2 Tests . . . 1

2 Data Structure Index 3

2.1 Data Structures . . . 3

3 Data Structure Documentation 5

3.1 Block Struct Reference . . . 5

3.1.1 Detailed Description . . . 6

3.2 BlockData Struct Reference . . . 6

3.2.1 Detailed Description . . . 6

3.3 ChunkBlockchain Struct Reference . . . 7

3.3.1 Detailed Description . . . 7

3.4 client_connection Struct Reference . . . 8

3.4.1 Detailed Description . . . 8

3.5 Neighbour Struct Reference . . . 8

3.5.1 Detailed Description . . . 8

3.6 Node Struct Reference . . . 8

3.6.1 Detailed Description . . . 9

3.7 Transaction Struct Reference . . . 9

3.7.1 Detailed Description . . . 9

3.8 TransactionData Struct Reference . . . 10

3.8.1 Detailed Description . . . 10

3.9 Wallet Struct Reference . . . 10

3.9.1 Detailed Description . . . 10

Index 11

Chapter 1

PEPITAS

C cryptocurrency.

1.1 CODING STYLE

1.1.1 Coding case

- *Functions*, *variables* and *filenames* must be written in `snake_case`.
- *Structures* must be written in `PascalCase`.
- *Constants* or *MACRO* must be written in `UPPER_SNAKE_CASE`.

1.1.2 Tests

Each function must be tested before **marked as done**. To create a test function, you must write it in the `test/` directory and call the file `filename_test.c` and its functions `functionname_test`. Note that the test file must be at the same relative place than his real function

exemple : if you want to test `init_server()` in the file `network/client.c`, you must write the test in `test/network/client_test.c` and call the test function `init_server_test()`

Chapter 2

Data Structure Index

2.1 Data Structures

Here are the data structures with brief descriptions:

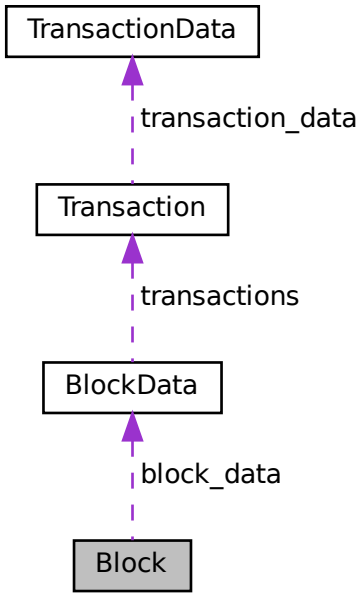
Block 5
BlockData 6
ChunkBlockchain 7
client_connection 8
Neighbour 8
Node 8
Transaction 9
TransactionData 10
Wallet 10

Chapter 3

Data Structure Documentation

3.1 Block Struct Reference

Collaboration diagram for Block:



Data Fields

- uint16_t **chunk_id**
- BlockData **block_data**
- size_t **signature_len**
- char * **block_signature**

3.1.1 Detailed Description

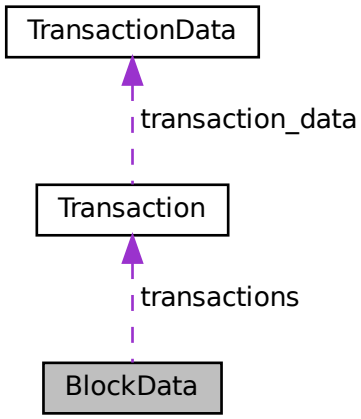
Definition at line 31 of file block.h.

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

- headers/core/blockchain/block.h

3.2 BlockData Struct Reference

Collaboration diagram for BlockData:



Data Fields

- char **magic**
- char **previous_block_hash** [SHA384_DIGEST_LENGTH ∗2+1]
- size_t **height**
- uint16_t **nb_transactions**
- Transaction ∗∗ **transactions**
- RSA ∗ **validator_public_key**
- time_t **block_timestamp**

3.2.1 Detailed Description

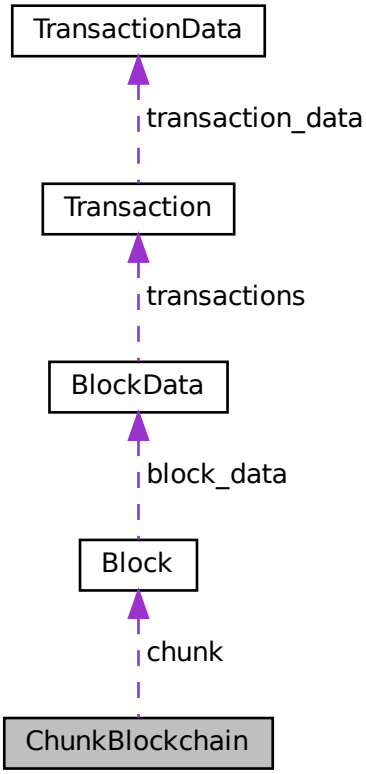
Definition at line 17 of file block.h.

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

- headers/core/blockchain/block.h

3.3 ChunkBlockchain Struct Reference

Collaboration diagram for ChunkBlockchain:



Data Fields

- size_t **chunk_nb**
- Block ** **chunk**

3.3.1 Detailed Description

Definition at line 41 of file block.h.

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

- headers/core/blockchain/block.h

3.4 client_connection Struct Reference

Data Fields

- struct addrinfo **info**
- int **socket**

3.4.1 Detailed Description

Definition at line 8 of file server.h.

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

- headers/network/server.h

3.5 Neighbour Struct Reference

Data Fields

- int **family**
- char * **hostname**
- int **server_sockfd**
- int **client_sockfd**

3.5.1 Detailed Description

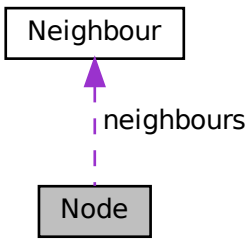
Definition at line 8 of file client.h.

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

- headers/network/client.h

3.6 Node Struct Reference

Collaboration diagram for Node:



Data Fields

- Neighbour * **neighbours**

3.6.1 Detailed Description

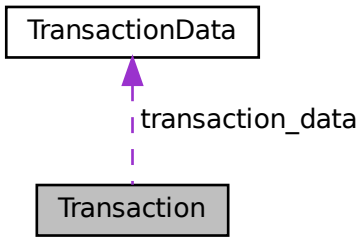
Definition at line 16 of file client.h.

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

- headers/network/client.h

3.7 Transaction Struct Reference

Collaboration diagram for Transaction:



Data Fields

- TransactionData * **transaction_data**
- size_t **signature_len**
- char * **transaction_signature**

3.7.1 Detailed Description

Definition at line 28 of file transaction.h.

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

- headers/core/blockchain/transaction.h

3.8 TransactionData Struct Reference

Data Fields

- RSA ∗ **sender_public_key**
- RSA ∗ **receiver_public_key**
- RSA ∗ **organisation_public_key**
- size_t **amount**
- size_t **sender_remaining_money**
- size_t **receiver_remaining_money**
- time_t **transaction_timestamp**
- char **cause** [512]
- char **asset** [512]

3.8.1 Detailed Description

Definition at line 11 of file transaction.h.

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

- headers/core/blockchain/transaction.h

3.9 Wallet Struct Reference

Data Fields

- RSA ∗ **priv_key**
- RSA ∗ **pub_key**
- size_t **amount**
- char **is_validator**

3.9.1 Detailed Description

Definition at line 10 of file wallet.h.

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

- headers/core/blockchain/wallet.h

Index

Block, 5
BlockData, 6

ChunkBlockchain, 7
client_connection, 8

Neighbour, 8
Node, 8

Transaction, 9
TransactionData, 10

Wallet, 10