

BlockchainCore

Создано системой Doxygen 1.9.1

1 Иерархический список классов	1
1.1 Иерархия классов	1
2 Алфавитный указатель классов	3
2.1 Классы	3
3 Классы	5
3.1 Класс BlockchainCore::Block	5
3.1.1 Подробное описание	7
3.1.2 Данные класса	7
3.1.2.1 SetCurHash	7
3.2 Структура BlockchainCore::BlockConsensusInfo	7
3.2.1 Подробное описание	8
3.3 Класс BlockchainCore::BlockContainedData	8
3.3.1 Подробное описание	8
3.4 Структура BlockchainCore::BlockHashInfo	9
3.4.1 Подробное описание	9
3.4.2 Методы	9
3.4.2.1 serialize()	9
3.5 Класс BlockchainCore::BlockLogs	10
3.6 Класс BlockchainCore::Crypto	10
3.7 Класс BlockchainCore::CryptoLogs_ImportPubKey	10
3.8 Класс BlockchainCore::CryptoLogs_TryToVerifyECDSA_CryptoPP	11
3.9 Класс BlockchainCore::DisabledLogger	11
3.9.1 Подробное описание	12
3.10 Класс BlockchainCore::ILogger	12
3.10.1 Подробное описание	13
3.11 Класс BlockchainCore::LoggerTester	13
3.11.1 Подробное описание	13
3.12 Структура BlockchainCore::LogPackage	13
3.12.1 Подробное описание	14
3.13 Класс BlockchainCore::TextOArchive_ForSign	14
3.13.1 Подробное описание	15
Предметный указатель	17

Глава 1

Иерархический список классов

1.1 Иерархия классов

Иерархия классов.

BlockchainCore::Block	5
BlockchainCore::BlockConsensusInfo	7
BlockchainCore::BlockContainedData	8
BlockchainCore::BlockHashInfo	9
BlockchainCore::BlockLogs	10
BlockchainCore::Crypto	10
BlockchainCore::CryptoLogs_ImportPubKey	10
BlockchainCore::CryptoLogs_TryToVerifyECDSA_CryptoPP	11
BlockchainCore::ILogger	12
BlockchainCore::DisabledLogger	11
BlockchainCore::LoggerTester	13
BlockchainCore::LogPackage	13
boost::archive::text_oarchive_impl	
BlockchainCore::TextOArchive_ForSign	14

Глава 2

Алфавитный указатель классов

2.1 Классы

Классы с их кратким описанием.

BlockChainCore::Block	
Класс, хранящий информацию о блоке	5
BlockChainCore::BlockConsensusInfo	
Информация, необходимая алгоритму консенсуса	7
BlockChainCore::BlockContainedData	
Класс для хранения хранящихся в блоке данных в произвольном формате	8
BlockChainCore::BlockHashInfo	
Информация о хэше блока	9
BlockChainCore::BlockLogs	10
BlockChainCore::Crypto	10
BlockChainCore::CryptoLogs_ImportPubKey	10
BlockChainCore::CryptoLogs_TryToVerifyECDSA_CryptoPP	11
BlockChainCore::DisabledLogger	
Логгер выключен, чилим	11
BlockChainCore::ILogger	12
BlockChainCore::LoggerTester	13
BlockChainCore::LogPackage	
Структура, характеризующая отдельную запись в логах	13
BlockChainCore::TextOArchive_ForSign	14

Глава 3

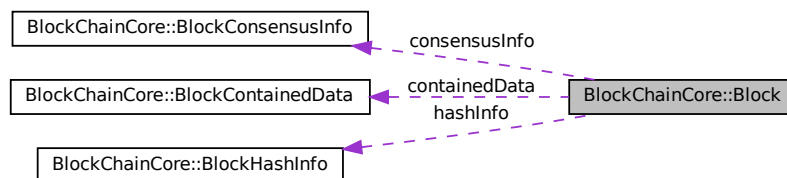
Классы

3.1 Класс BlockchainCore::Block

Класс, хранящий информацию о блоке

```
#include <Block.h>
```

Граф связей класса BlockchainCore::Block:



Открытые члены

- Block (const [BlockHashInfo](#) &hashInfo, const UnixTime ×tamp, const std::pair< std::string, std::string > &minedBy, const BigNums::mpz_int &ledgerId, const [BlockConsensusInfo](#) &consensusInfo, const [BlockContainedData](#) &containedData)
- Block ([BlockHashInfo](#) &&hashInfo, const UnixTime ×tamp, std::pair< std::string, std::string > &&minedBy, const BigNums::mpz_int &ledgerId, const [BlockConsensusInfo](#) &consensusInfo, [BlockContainedData](#) &&containedData)
- auto GetHashInfo () const noexcept
- ByteVector [SerializeForHashing](#) () const
 - Осуществляет сериализацию блока для подсчета его хеша.
- ByteVector GetBlockBytes () const
- void SetCurHash (ByteVector &&curHash)
- void SetPrevHash (const ByteVector &prevHash)
- void SetPrevHash (ByteVector &&prevHash)
- void SetTransactionId (const BigNums::mpz_int &ledgerId)
- void SetTransactionId (BigNums::mpz_int &&ledgerId)

- `auto GetTimestamp () const noexcept`
- `void SetTimestamp (const UnixTime ×tamp)`
- `auto GetMinedBy () const noexcept`
- `void SetMinedBy (const std::pair< std::string, std::string > &minedBy)`
- `void SetMinedBy (std::pair< std::string, std::string > &&minedBy)`
- `auto GetConsensusInfo () const noexcept`
- `void SetLuck (const BigNums::mpq_rational &luck)`
- `void SetMiningPoint (const BigNums::mpz_int &miningPoint)`
- `auto GetContainedData () const noexcept`
- `void SetContainedData (const BlockContainedData &contData)`
- `void SetContainedData (BlockContainedData &&contData)`
- `bool TryWriteToStream (std::ostream &outStream) noexcept`

Открытые статические члены

- `static Block ConstructFromChain (const BlockChain ¤tBlockChain) noexcept(false)`
Частично конструирует блок, цепляя его к переданной ветке блокчейна
- `static Block init ()`
Инициализирует начальный блок.
- `static std::optional< Block > GetFromStream (std::istream &inputStream) noexcept`

Открытые атрибуты

- `template<class CallableVerifier >`
`bool IsValid(CallableVerifier &&callable) requires std void SetCurHash (const ByteVector &cur← Hash)`

Закрытые члены

- `Block (const BlockChain ¤tBlockChain) noexcept(false)`
- `template<typename Archive >`
`void save (Archive &ar, const unsigned int version) const`
- `template<class Archive >`
`void load (Archive &ar, const unsigned int version)`
- `BOOST_SERIALIZATION_SPLIT_MEMBER ()`

Закрытые данные

- `BlockHashInfo hashInfo`
- `UnixTime timestamp`
- `std::pair< std::string, std::string > minedBy`
- `BigNums::mpz_int ledgerId = 1`
- `BlockConsensusInfo consensusInfo`
- `BlockContainedData containedData`

Друзья

- `class boost::serialization::access`

3.1.1 Подробное описание

Класс, хранящий информацию о блоке

Класс, хранящий информацию о блоке. По сути, его основная задача - обеспечение сериализации данных.

3.1.2 Данные класса

3.1.2.1 SetCurHash

```
template<class CallableVerifier >  
void BlockchainCore::Block::SetCurHash
```

Шаблонная функция, осуществляющая сравнение посчитанного хеша с текущими данными

Объявления и описания членов классов находятся в файлах:

- /home/houdini/development/Cpp/PoWBlockchain/src/BlockchainCore/src/Block/Block.h
- /home/houdini/development/Cpp/PoWBlockchain/src/BlockchainCore/src/Block/Block.cpp

3.2 Структура BlockchainCore::BlockConsensusInfo

Информация, необходимая алгоритму консенсуса

```
#include <Block.h>
```

Открытые атрибуты

- BigNums::mpz_int miningPoint = 0
- BigNums::mpq_rational luck = 0.0

Закрытые члены

- template<typename Archive >
void save (Archive &ar, const unsigned int version) const
- template<class Archive >
void load (Archive &ar, const unsigned int version)
- BOOST_SERIALIZATION_SPLIT_MEMBER ()

Друзья

- class boost::serialization::access

3.2.1 Подробное описание

Информация, необходимая алгоритму консенсуса

Объявления и описания членов структуры находятся в файле:

- `/home/houdini/development/Cpp/PoWBlockchain/src/BlockChainCore/src/Block/Block.h`

3.3 Класс `BlockChainCore::BlockContainedData`

Класс для хранения хранящихся в блоке данных в произвольном формате.

```
#include <BlockContainedData.h>
```

Открытые члены

- `ByteVector & GetData () noexcept`

Закрытые члены

- `template<typename Archive >
void serialize (Archive &ar, const unsigned int version)`

Закрытые данные

- `ByteVector data`

Друзья

- `class boost::serialization::access`

3.3.1 Подробное описание

Класс для хранения хранящихся в блоке данных в произвольном формате.

Единственное, за что отвечает класс `BlockContainedData` - это хранения и сериализация данных. Данные, хранимые в нем, абсолютно произвольны. За валидацию хранимых данных отвечают внешние ресурсы, никакого отношения к ядру абсолютно не имеющие.

Объявления и описания членов классов находятся в файлах:

- `/home/houdini/development/Cpp/PoWBlockchain/src/BlockChainCore/src/Block/Block↵
ContainedData.h`
- `/home/houdini/development/Cpp/PoWBlockchain/src/BlockChainCore/src/Block/Block↵
ContainedData.cpp`

3.4 Структура BlockChainCore::BlockHashInfo

Информация о хэше блока

```
#include <Block.h>
```

Открытые атрибуты

- ByteVector prevSignedHash
- ByteVector curSignedHash

Закрытые члены

- template<typename Archive >
void [serialize](#) (Archive &ar, const unsigned int version)
- template<> void [serialize](#) ([TextOArchive_ForSign](#) &ar, const unsigned int version)

Друзья

- class boost::serialization::access

3.4.1 Подробное описание

Информация о хэше блока

Информация о хэше блока. Хранит подписанный хэш предыдущей и текущей записи.

3.4.2 Методы

3.4.2.1 `serialize()`

```
template<typename Archive >  
void BlockChainCore::BlockHashInfo::serialize (  
    Archive & ar,  
    const unsigned int version ) [inline], [private]
```

Версия сериализатора, сохраняющая всю информацию о хэше блока, в том числе и о текущем подписанном хэше. Используется для последующего хранения данных о блоке.

Объявления и описания членов структур находятся в файлах:

- /home/houdini/development/Cpp/PoWBlockchain/src/BlockChainCore/src/Block/Block.h
- /home/houdini/development/Cpp/PoWBlockchain/src/BlockChainCore/src/Block/Block.cpp

3.5 Класс BlockchainCore::BlockLogs

Открытые статические члены

- static void LogIsValidInfo (const boost::multiprecision::mpz_int &ledgerId, const std::source_location &loc=std::source_location::current())
- template<ExceptionConcept ExceptionType>
static void LogBlockException (ExceptionType &ex, std::string_view exType, const std::source_location &loc=std::source_location::current())
- static void LogBlockUnknownException (std::string_view message, const std::source_location &loc=std::source_location::current())

Объявления и описания членов классов находятся в файлах:

- /home/houdini/development/Cpp/PoWBlockchain/src/BlockchainCore/src/Block/BlockLogs.h
- /home/houdini/development/Cpp/PoWBlockchain/src/BlockchainCore/src/Block/BlockLogs.cpp

3.6 Класс BlockchainCore::Crypto

Открытые типы

- using ByteVector = std::vector< unsigned char >

Открытые статические члены

- static bool TryToVerifyECDSA_CryptoPP (const ByteVector &signature, const ByteVector &blockData, const std::pair< std::string, std::string > &publicKey) noexcept

Объявления и описания членов классов находятся в файлах:

- /home/houdini/development/Cpp/PoWBlockchain/src/BlockchainCore/src/Crypto/Crypto.h
- /home/houdini/development/Cpp/PoWBlockchain/src/BlockchainCore/src/Crypto/Crypto.cpp

3.7 Класс BlockchainCore::CryptoLogs_ImportPubKey

Открытые статические члены

- static void LogStringStreamFail (const std::pair< std::string, std::string > &publicKey, const std::source_location &loc=std::source_location::current())
- static void LogValidatingFail (const std::pair< std::string, std::string > &publicKey, const std::source_location &loc=std::source_location::current())
- static void LogCaughtUnknownException (const std::pair< std::string, std::string > &publicKey, const std::source_location &loc=std::source_location::current())

Объявления и описания членов классов находятся в файлах:

- /home/houdini/development/Cpp/PoWBlockchain/src/BlockchainCore/src/Crypto/CryptoLogs_ImportPubKey.h
- /home/houdini/development/Cpp/PoWBlockchain/src/BlockchainCore/src/Crypto/CryptoLogs_ImportPubKey.cpp

3.8 Класс BlockchainCore::CryptoLogs_TryToVerifyECDSA_CryptoPP

Открытые типы

- using ByteVector = std::vector< unsigned char >

Открытые статические члены

- static void LogVerifyMessageFail (const ByteVector &signature, const ByteVector &blockData, const std::pair< std::string, std::string > &publicKey, const std::source_location &loc=std::source_location::current())
- static void LogUnknownExceptionCaught (const ByteVector &signature, const ByteVector &blockData, const std::pair< std::string, std::string > &publicKey, const std::source_location &loc=std::source_location::current())

Объявления и описания членов классов находятся в файлах:

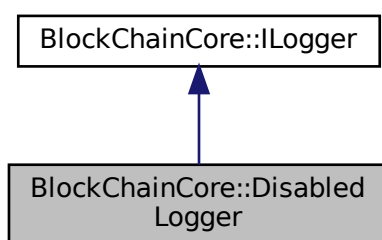
- /home/houdini/development/Cpp/PoWBlockchain/src/BlockchainCore/src/Crypto/Crypto↔Logs_ImportPubKey.h
- /home/houdini/development/Cpp/PoWBlockchain/src/BlockchainCore/src/Crypto/Crypto↔Logs_ImportPubKey.cpp

3.9 Класс BlockchainCore::DisabledLogger

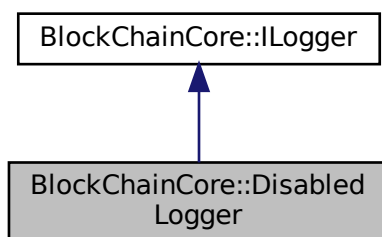
Логгер выключен, чилим.

```
#include <ILogger.h>
```

Граф наследования:BlockchainCore::DisabledLogger:



Граф связей класса BlockchainCore::DisabledLogger:



Открытые члены

- void Log (const [LogPackage](#) &log) override

3.9.1 Подробное описание

Логгер выключен, чилим.

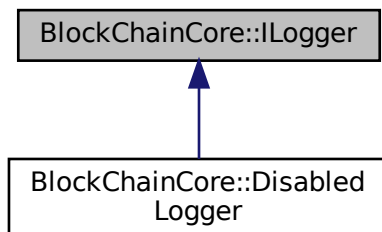
Объявления и описания членов классов находятся в файлах:

- /home/houdini/development/Cpp/PoWBlockchain/src/BlockchainCore/src/Logger/ILogger.h
- /home/houdini/development/Cpp/PoWBlockchain/src/BlockchainCore/src/Logger/ILogger.cpp

3.10 Класс BlockchainCore::ILogger

```
#include <ILogger.h>
```

Граф наследования:BlockchainCore::ILogger:



Открытые члены

- virtual void Log (const [LogPackage](#) &log)=0

3.10.1 Подробное описание

Интерфейс, с помощью которого осуществляется логирование. Важно! Так как существуют различные виды логгеров по наличию или отсутствию перехвата ошибок и прочего, данный интерфейс отвечает лишь за доставку данных до логгера. Следует учитывать, что экземпляр данного класса будет создан на самых ранних стадиях начала функционирования ядра и в дальнейшем редактированию не подлежит. Со всеми аварийными ситуациями должен справляться сам механизм логирования.

Важно! Под капотом используется shared_ptr. В общем, нужно грамотно определить, как деструктор влияет на время жизни объекта. (например, ресурсами объекта владеет го-шный логгер)

Для обеспечения потокобезопасности Logger не должен менять свои состояния во время работы, или же должен быть защищен примитивами синхронизации.

Объявления и описания членов класса находятся в файле:

- /home/houdini/development/Cpp/PoWBlockchain/src/BlockChainCore/src/Logger/ILogger.h

3.11 Класс BlockChainCore::LoggerTester

```
#include <ILogger.h>
```

Открытые члены

- LoggerTester (std::shared_ptr< [ILogger](#) > logger)

3.11.1 Подробное описание

Класс для тестирования логгера. Использует RAII, подсовывая переданный логгер в Singleton и возвращая [DisabledLogger](#) после уничтожения. Так как используется синглтон, следует быть аккуратным при тестах при запуске тестов в параллельном режиме.

Объявления и описания членов классов находятся в файлах:

- /home/houdini/development/Cpp/PoWBlockchain/src/BlockChainCore/src/Logger/ILogger.h
- /home/houdini/development/Cpp/PoWBlockchain/src/BlockChainCore/src/Logger/ILogger.cpp

3.12 Структура BlockChainCore::LogPackage

Структура, характеризующая отдельную запись в логах.

```
#include <ILogger.h>
```

Открытые атрибуты

- LogTypeEnum logType
- std::string_view systemInfo
- boost::posix_time::ptime timestamp
- std::string_view message
- std::string_view context
- std::string_view sourceType

3.12.1 Подробное описание

Структура, характеризующая отдельную запись в логах.

Структура [LogPackage](#) содержит следующий набор полей:

logType - отвечает за уровень записи

systemInfo - любая дополнительная информация о системе (timestamp вынесен отдельно)

timestamp - временная метка в формате Posix. Характеризует время, в которое был сгенерирован лог

message - сообщение об ошибке

context - любой дополнительный контекст об ошибке

sourceType - тип источника в формате строки. Самое щекотливое с точки зрения формата поле. По умолчанию все логи от ядра генерируются с sourceType BlockchainCore. Возможно, что данное поле вообще будет игнорироваться. Однако, так как ваш логгер может работать не только с логами ядра, а ядро вообще ничего не должно знать о том, куда и как оно логирует, данное поле было добавлено для возможности дополнительной кастомизации маршрутизации логов

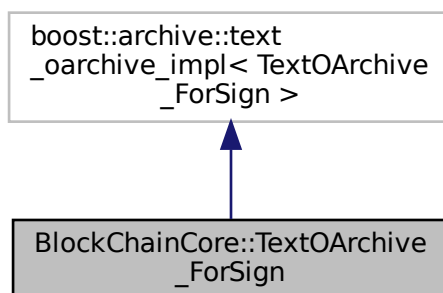
Объявления и описания членов структуры находятся в файле:

- /home/houdini/development/Cpp/PoWBlockchain/src/BlockchainCore/src/Logger/ILogger.h

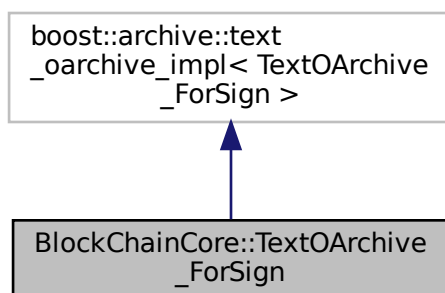
3.13 Класс BlockchainCore::TextOArchive_ForSign

```
#include <Block.h>
```

Граф наследования:BlockchainCore::TextOArchive_ForSign:



Граф связей класса BlockchainCore::TextOArchive_ForSign:



Открытые члены

- `TextOArchive_ForSign (std::ostream &os_, unsigned int flags=0)`

3.13.1 Подробное описание

Данный класс полностью повторяет функционал класса `boost::archive::text_oarchive`. Его единственная задача - обеспечить дополнительный сценарий сериализации для верификации блока. Достигается это путем перегрузки шаблона `serialize` в структуре [BlockHashInfo](#).

Объявления и описания членов класса находятся в файле:

- `/home/houdini/development/Cpp/PoWBlockchain/src/BlockchainCore/src/Block/Block.h`

Предметный указатель

BlockchainCore::Block, [5](#)
 SetCurHash, [7](#)
BlockchainCore::BlockConsensusInfo, [7](#)
BlockchainCore::BlockContainedData, [8](#)
BlockchainCore::BlockHashInfo, [9](#)
 serialize, [9](#)
BlockchainCore::BlockLogs, [10](#)
BlockchainCore::Crypto, [10](#)
BlockchainCore::CryptoLogs_ImportPubKey, [10](#)
BlockchainCore::CryptoLogs_TryToVerifyECDSA_CryptoPP,
 [11](#)
BlockchainCore::DisabledLogger, [11](#)
BlockchainCore::ILogger, [12](#)
BlockchainCore::LoggerTester, [13](#)
BlockchainCore::LogPackage, [13](#)
BlockchainCore::TextOArchive_ForSign, [14](#)

serialize
 BlockchainCore::BlockHashInfo, [9](#)
SetCurHash
 BlockchainCore::Block, [7](#)