

VServer

Создано системой Doxygen 1.9.4

1	Алфавитный указатель пространств имен	1
1.1	Пространства имен	1
2	Иерархический список классов	3
2.1	Иерархия классов	3
3	Алфавитный указатель классов	5
3.1	Классы	5
4	Список файлов	7
4.1	Файлы	7
5	Пространства имен	9
5.1	Пространство имен <code>crutils</code>	9
5.1.1	Подробное описание	9
5.1.2	Функции	9
5.1.2.1	<code>calculateHash()</code>	9
6	Классы	11
6.1	Класс <code>ArgsDecodeException</code>	11
6.2	Класс <code>AuthException</code>	11
6.2.1	Подробное описание	11
6.3	Класс <code>DataDecodeException</code>	12
6.3.1	Подробное описание	12
6.4	Класс <code>Exception</code>	12
6.4.1	Подробное описание	13
6.4.2	Конструктор(ы)	13
6.4.2.1	<code>Exception()</code>	13
6.4.3	Методы	13
6.4.3.1	<code>getFunc()</code>	13
6.4.3.2	<code>getMessage()</code>	14
6.4.3.3	<code>getName()</code>	14
6.4.3.4	<code>getTime()</code>	14
6.4.3.5	<code>isCritical()</code>	14
6.4.3.6	<code>what()</code>	15
6.5	Класс <code>Interface</code>	15
6.5.1	Подробное описание	15
6.5.2	Конструктор(ы)	15
6.5.2.1	<code>Interface()</code>	15
6.5.3	Методы	16
6.5.3.1	<code>getAddress()</code>	16
6.5.3.2	<code>getPathToDB()</code>	16
6.5.3.3	<code>getPathToLog()</code>	16
6.5.3.4	<code>getPort()</code>	17
6.6	Класс <code>Network</code>	17

6.6.1	Подробное описание	17
6.6.2	Конструктор(ы)	18
6.6.2.1	Network()	18
6.6.3	Методы	18
6.6.3.1	auth()	18
6.6.3.2	calc()	18
6.6.3.3	getAddress()	19
6.6.3.4	getDatabase()	19
6.6.3.5	getPort()	19
6.6.3.6	wait()	19
6.7	Класс NetworkException	20
6.7.1	Подробное описание	20
6.8	Класс RWException	20
6.8.1	Подробное описание	20
7	Файлы	21
7.1	Файл /home/student/Документы/busigin/server/source/modules/network.h	21
7.1.1	Подробное описание	21
7.2	network.h	22
7.3	Файл /home/student/Документы/busigin/server/source/modules/exceptions.h	22
7.3.1	Подробное описание	23
7.4	exceptions.h	23
7.5	Файл /home/student/Документы/busigin/server/source/modules/crutils.h	24
7.5.1	Подробное описание	24
7.6	crutils.h	24
7.7	Файл /home/student/Документы/busigin/server/source/modules/rw.h	25
7.7.1	Подробное описание	25
7.7.2	Функции	25
7.7.2.1	readDataBase()	25
7.7.2.2	writeLog()	26
7.8	rw.h	26
7.9	Файл /home/student/Документы/busigin/server/source/modules/interface.h	27
7.9.1	Подробное описание	27
7.10	interface.h	27
7.11	Файл /home/student/Документы/busigin/server/source/main.cpp	28
7.11.1	Подробное описание	28
7.11.2	Функции	28
7.11.2.1	main()	29
7.11.2.2	run()	30
	Предметный указатель	31

Глава 1

Алфавитный указатель пространств имен

1.1 Пространства имен

Полный список документированных пространств имен.

crutils	
Пространство имен для криптографических утилит	9

Глава 2

Иерархический список классов

2.1 Иерархия классов

Иерархия классов.

std::exception	
Exception	12
ArgsDecodeException	11
AuthException	11
DataDecodeException	12
NetworkException	20
RWException	20
Interface	15
Network	17

Глава 3

Алфавитный указатель классов

3.1 Классы

Классы с их кратким описанием.

ArgsDecodeException	
Класс для исключений при декодировании аргументов	11
AuthException	
Класс для исключений аутентификации	11
DataDecodeException	
Класс для исключений при декодировании данных	12
Exception	
Базовый класс для исключений	12
Interface	
Класс для управления интерфейсом программы	15
Network	
Класс для управления сетевым подключением и взаимодействием	17
NetworkException	
Класс для сетевых исключений	20
RWException	
Класс для исключений ввода-вывода	20

Глава 4

Список файлов

4.1 Файлы

Полный список документированных файлов.

/home/student/Документы/busigin/server/source/modules/ crutils.h	
Определения вспомогательных функций для криптографических операций	24
/home/student/Документы/busigin/server/source/modules/ exceptions.h	
Определение классов исключений	22
/home/student/Документы/busigin/server/source/modules/ interface.h	
Определение класса интерфейса	27
/home/student/Документы/busigin/server/source/modules/ network.h	
Определения классов для управления сетевым взаимодействием	21
/home/student/Документы/busigin/server/source/modules/ rw.h	
Определение функций для работы с файлами	25
/home/student/Документы/busigin/server/source/ main.cpp	
Главный файл программы	28

Глава 5

Пространства имен

5.1 Пространство имен crutils

Пространство имен для криптографических утилит.

Функции

- `std::string calculateHash (const std::string &data)`
Функция для вычисления хеша.

5.1.1 Подробное описание

Пространство имен для криптографических утилит.

5.1.2 Функции

5.1.2.1 calculateHash()

```
std::string crutils::calculateHash (  
    const std::string & data )
```

Функция для вычисления хеша.

Аргументы

data	Данные для хеширования.
------	-------------------------

Возвращает

Хеш в виде строки.

Глава 6

Классы

6.1 Класс ArgsDecodeException

Класс для исключений при декодировании аргументов.

```
#include <exceptions.h>
```

Граф наследования: ArgsDecodeException:

6.2 Класс AuthException

Класс для исключений аутентификации.

```
#include <exceptions.h>
```

Граф наследования: AuthException:

Граф связей класса AuthException:

Открытые члены

- AuthException (const std::string &message, const std::string &func="Undefended", bool is_critical=false)

Дополнительные унаследованные члены

6.2.1 Подробное описание

Класс для исключений аутентификации.

Объявления и описания членов классов находятся в файлах:

- /home/student/Документы/busigin/server/source/modules/exceptions.h
- /home/student/Документы/busigin/server/source/modules/exceptions.cpp

6.3 Класс DataDecodeException

Класс для исключений при декодировании данных.

```
#include <exceptions.h>
```

Граф наследования:DataDecodeException:

Граф связей класса DataDecodeException:

Открытые члены

- DataDecodeException (const std::string &message, const std::string &func="Undefended", bool is_critical=false)

Дополнительные унаследованные члены

6.3.1 Подробное описание

Класс для исключений при декодировании данных.

Объявления и описания членов классов находятся в файлах:

- /home/student/Документы/busigin/server/source/modules/exceptions.h
- /home/student/Документы/busigin/server/source/modules/exceptions.cpp

6.4 Класс Exception

Базовый класс для исключений.

```
#include <exceptions.h>
```

Граф наследования:Exception:

Граф связей класса Exception:

Открытые члены

- Exception (const std::string &message, const std::string &func="Undefended", bool is_critical=false)
Конструктор класса Exception.
- std::string getName () const
Получение имени исключения.
- std::string getFunc () const
Получение имени функции, в которой возникло исключение.
- std::string getMessage () const
Получение сообщения об ошибке.
- std::string getTime () const
Получение времени возникновения исключения.
- bool isCritical () const
Проверка флага критичности исключения.
- const char * what () const noexcept override
Метод для получения сообщения об ошибке.

Защищенные данные

- `std::string name`
Имя исключения.
- `std::string func`
Имя функции, в которой возникло исключение.
- `std::string message`
Сообщение об ошибке.
- `std::string time`
Время возникновения исключения.
- `bool is_critical`
Флаг критичности исключения.
- `std::string what_message`
Полное сообщение об ошибке.

6.4.1 Подробное описание

Базовый класс для исключений.

6.4.2 Конструктор(ы)

6.4.2.1 Exception()

```
Exception::Exception (
    const std::string & message,
    const std::string & func = "Undefended",
    bool is_critical = false )
```

Конструктор класса [Exception](#).

Аргументы

message	Сообщение об ошибке.
func	Имя функции, в которой возникло исключение (по умолчанию "Undefended").
is_critical	Флаг критичности исключения (по умолчанию false).

6.4.3 Методы

6.4.3.1 getFunc()

```
std::string Exception::getFunc ( ) const
```

Получение имени функции, в которой возникло исключение.

Возвращает

Имя функции.

6.4.3.2 getMessage()

```
std::string Exception::getMessage ( ) const
```

Получение сообщения об ошибке.

Возвращает

Сообщение об ошибке.

6.4.3.3 getName()

```
std::string Exception::getName ( ) const
```

Получение имени исключения.

Возвращает

Имя исключения.

6.4.3.4 getTime()

```
std::string Exception::getTime ( ) const
```

Получение времени возникновения исключения.

Возвращает

Время возникновения исключения.

6.4.3.5 isCritical()

```
bool Exception::isCritical ( ) const
```

Проверка флага критичности исключения.

Возвращает

true, если исключение критическое, иначе false.

6.4.3.6 what()

```
const char * Exception::what ( ) const [override], [noexcept]
```

Метод для получения сообщения об ошибке.

Возвращает

Сообщение об ошибке.

Объявления и описания членов классов находятся в файлах:

- /home/student/Документы/busigin/server/source/modules/[exceptions.h](#)
- /home/student/Документы/busigin/server/source/modules/exceptions.cpp

6.5 Класс Interface

Класс для управления интерфейсом программы.

```
#include <interface.h>
```

Открытые члены

- [Interface](#) ()
Конструктор принимает параметры командной строки.
- void parseArgs (int argc, char *argv[])
Метод для вывода справки.
- void showHelp () const
- std::string & [getPathToDB](#) ()
Метод для получения пути к файлу базы данных.
- std::string & [getPathToLog](#) ()
Метод для получения пути к файлу логов.
- std::string & [getAddress](#) ()
Метод для получения адреса.
- int [getPort](#) () const
Метод для получения порта.

6.5.1 Подробное описание

Класс для управления интерфейсом программы.

6.5.2 Конструктор(ы)

6.5.2.1 Interface()

```
Interface::Interface ( )
```

Конструктор принимает параметры командной строки.

Аргументы

argc	Количество аргументов командной строки.
argv	Аргументы командной строки.

Исключения

ArgsDecodeException	Если аргументы командной строки некорректны.
-------------------------------------	--

6.5.3 Методы

6.5.3.1 getAddress()

std::string & Interface::getAddress ()

Метод для получения адреса.

Возвращает

Адрес.

6.5.3.2 getPathToDB()

std::string & Interface::getPathToDB ()

Метод для получения пути к файлу базы данных.

Возвращает

Путь к файлу базы данных.

6.5.3.3 getPathToLog()

std::string & Interface::getPathToLog ()

Метод для получения пути к файлу логов.

Возвращает

Путь к файлу логов.

6.5.3.4 getPort()

```
int Interface::getPort ( ) const
```

Метод для получения порта.

Возвращает

Порт.

Объявления и описания членов классов находятся в файлах:

- /home/student/Документы/busigin/server/source/modules/[interface.h](#)
- /home/student/Документы/busigin/server/source/modules/interface.cpp

6.6 Класс Network

Класс для управления сетевым подключением и взаимодействием.

```
#include <network.h>
```

Открытые члены

- [Network](#) (const std::string &address, uint16_t port, const std::map< std::string, std::string > &database)
Конструктор класса [Network](#).
- std::string & [getAddress](#) ()
Метод для получения адреса сервера.
- uint16_t & [getPort](#) ()
Метод для получения порта сервера.
- std::map< std::string, std::string > & [getDatabase](#) ()
Метод для получения базы данных пользователей.
- void [wait](#) ()
Метод для ожидания соединений.
- void [auth](#) ()
Метод для аутентификации пользователя.
- void [calc](#) ()
Метод для вычисления суммы значений в векторах.
- void close ()
Метод для закрытия клиентского соединения.
- void quit ()
Метод для закрытия основного соединения.

6.6.1 Подробное описание

Класс для управления сетевым подключением и взаимодействием.

6.6.2 Конструктор(ы)

6.6.2.1 Network()

```
Network::Network (
    const std::string & address,
    uint16_t port,
    const std::map< std::string, std::string > & database )
```

Конструктор класса [Network](#).

Аргументы

address	Адрес сервера.
port	Порт сервера.
database	База данных пользователей.

6.6.3 Методы

6.6.3.1 auth()

```
void Network::auth ( )
```

Метод для аутентификации пользователя.

Исключения

NetworkException	Если не удалось прочитать логин, найти логин в базе данных, отправить или получить данные от клиента.
----------------------------------	---

6.6.3.2 calc()

```
void Network::calc ( )
```

Метод для вычисления суммы значений в векторах.

Исключения

NetworkException	Если не удалось прочитать данные от клиента или отправить результат.
----------------------------------	--

6.6.3.3 getAddress()

```
std::string & Network::getAddress ( )
```

Метод для получения адреса сервера.

Возвращает

Адрес сервера.

6.6.3.4 getDatabase()

```
std::map< std::string, std::string > & Network::getDatabase ( )
```

Метод для получения базы данных пользователей.

Возвращает

База данных пользователей.

6.6.3.5 getPort()

```
uint16_t & Network::getPort ( )
```

Метод для получения порта сервера.

Возвращает

Порт сервера.

6.6.3.6 wait()

```
void Network::wait ( )
```

Метод для ожидания соединений.

Исключения

NetworkException	Если не удалось создать, привязать или слушать сокет.
----------------------------------	---

Объявления и описания членов классов находятся в файлах:

- `/home/student/Документы/busigin/server/source/modules/network.h`
- `/home/student/Документы/busigin/server/source/modules/network.cpp`

6.7 Класс NetworkException

Класс для сетевых исключений.

```
#include <exceptions.h>
```

Граф наследования: NetworkException:

Граф связей класса NetworkException:

Открытые члены

- `NetworkException (const std::string &message, const std::string &func="Undefended", bool is_critical=false)`

Дополнительные унаследованные члены

6.7.1 Подробное описание

Класс для сетевых исключений.

Объявления и описания членов классов находятся в файлах:

- `/home/student/Документы/busigin/server/source/modules/exceptions.h`
- `/home/student/Документы/busigin/server/source/modules/exceptions.cpp`

6.8 Класс RWException

Класс для исключений ввода-вывода.

```
#include <exceptions.h>
```

Граф наследования: RWException:

Граф связей класса RWException:

Открытые члены

- `RWException (const std::string &message, const std::string &func="Undefended", bool is_critical=false)`

Дополнительные унаследованные члены

6.8.1 Подробное описание

Класс для исключений ввода-вывода.

Объявления и описания членов классов находятся в файлах:

- `/home/student/Документы/busigin/server/source/modules/exceptions.h`
- `/home/student/Документы/busigin/server/source/modules/exceptions.cpp`

Глава 7

Файлы

7.1 Файл

/home/student/Документы/busigin/server/source/modules/network.h

Определения классов для управления сетевым взаимодействием.

```
#include <map>
#include <string>
#include "crutils.h"
#include "exceptions.h"
```

Граф включаемых заголовочных файлов для network.h: Граф файлов, в которые включается этот файл:

Классы

- class [Network](#)

Класс для управления сетевым подключением и взаимодействием.

7.1.1 Подробное описание

Определения классов для управления сетевым взаимодействием.

Этот файл содержит определения классов для управления сетевыми подключениями и передачей данных.

Дата

23.11.2024

Версия

1.0

Автор

Бусыгин Г. С.

7.2 network.h

См. документацию.

```
1 #ifndef NETWORK_H
2 #define NETWORK_H
3
4 #include <map>
5 #include <string>
6 #include "crutils.h"
7 #include "exceptions.h"
8
9 class Network
10 {
11 public:
12     Network(
13         const std::string &address,
14         uint16_t port,
15         const std::map<std::string, std::string> &database);
16
17     std::string &getAddress();
18
19     uint16_t &getPort();
20
21     std::map<std::string, std::string> &getDatabase();
22
23     void wait();
24
25     void auth();
26
27     void calc();
28
29     void close();
30
31     void quit();
32 private:
33     std::string address;
34     uint16_t port;
35     std::map<std::string, std::string> database;
36     int socket;
37     int client_socket;
38 };
39 #endif // NETWORK_H
```

7.3 Файл

/home/student/Документы/busigin/server/source/modules/exceptions.h

Определение классов исключений.

```
#include <exception>
#include <string>
```

Граф включаемых заголовочных файлов для exceptions.h: Граф файлов, в которые включается этот файл:

Классы

- class [Exception](#)
Базовый класс для исключений.
- class [RWException](#)
Класс для исключений ввода-вывода.
- class [ArgsDecodeException](#)
Класс для исключений при декодировании аргументов.
- class [DataDecodeException](#)
Класс для исключений при декодировании данных.
- class [AuthException](#)
Класс для исключений аутентификации.
- class [NetworkException](#)
Класс для сетевых исключений.

7.3.1 Подробное описание

Определение классов исключений.

Этот файл содержит определения классов исключений для обработки различных ошибок в программе.

Дата

23.11.2024

Версия

1.0

Автор

Бусыгин Г. С.

7.4 exceptions.h

[См. документацию.](#)

```

1 #ifndef EXCEPTIONS_H
2 #define EXCEPTIONS_H
3
4 #include <exception>
5 #include <string>
6
19 class Exception : public std::exception {
20 public:
27     Exception(const std::string &message, const std::string &func = "Undefended", bool is_critical = false);
28
29     std::string getName() const;
30     std::string getFunc() const;
31     std::string getMessage() const;
32     std::string getTime() const;
33     bool isCritical() const;
34
39     const char *what() const noexcept override;
40
41 protected:
42     std::string name;
43     std::string func;
44     std::string message;
45     std::string time;
46     bool is_critical;
47     mutable std::string what_message;
48 };
49
53 class RWException : public Exception {
54 public:
55     RWException(const std::string &message, const std::string &func = "Undefended", bool is_critical = false);
56 };
57
61 class ArgsDecodeException : public Exception {
62 public:
63     ArgsDecodeException(const std::string &message, const std::string &func = "Undefended", bool is_critical = false);
64 };
65
69 class DataDecodeException : public Exception {
70 public:
71     DataDecodeException(const std::string &message, const std::string &func = "Undefended", bool is_critical = false);
72 };
73
77 class AuthException : public Exception {
78 public:
79     AuthException(const std::string &message, const std::string &func = "Undefended", bool is_critical = false);
80 };
81
85 class NetworkException : public Exception {
86 public:
87     NetworkException(const std::string &message, const std::string &func = "Undefended", bool is_critical = false);
88 };
89
90 #endif // EXCEPTIONS_H

```

7.5 Файл

/home/student/Документы/busigin/server/source/modules/crutils.h

Определения вспомогательных функций для криптографических операций.

```
#include <string>
```

Граф включаемых заголовочных файлов для crutils.h: Граф файлов, в которые включается этот файл:

Пространства имен

- namespace `crutils`

Пространство имен для криптографических утилит.

Функции

- `std::string crutils::calculateHash` (const std::string &data)

Функция для вычисления хеша.

7.5.1 Подробное описание

Определения вспомогательных функций для криптографических операций.

Этот файл содержит определения функций для генерации соли и вычисления хеша.

Дата

23.11.2024

Версия

1.0

Автор

Бусыгин Г. С.

7.6 crutils.h

[См. документацию.](#)

```
1 #ifndef CRUTILS_H
2 #define CRUTILS_H
3
4 #include <string>
5
6 namespace crutils {
7     std::string calculateHash(const std::string &data);
8 }
9
10 #endif // CRUTILS_H
```

7.7 Файл

/home/student/Документы/busigin/server/source/modules/rw.h

Определение функций для работы с файлами.

```
#include <map>
#include <string>
#include "exceptions.h"
```

Граф включаемых заголовочных файлов для rw.h: Граф файлов, в которые включается этот файл:

Функции

- `std::map< std::string, std::string > rw::readDataBase (const std::string &path)`
Функция для чтения базы данных.
- `void rw::writeLog (const std::string &path, const Exception &e)`
Функция для записи лога.

7.7.1 Подробное описание

Определение функций для работы с файлами.

Этот файл содержит определения функций для чтения базы данных и записи логов.

Дата

23.11.2024

Версия

1.0

Автор

Бусыгин Г. С.

7.7.2 Функции

7.7.2.1 readDataBase()

```
std::map< std::string, std::string > rw::readDataBase (  
    const std::string & path )
```

Функция для чтения базы данных.

Аргументы

path	Путь к файлу базы данных.
------	---------------------------

Возвращает

Возвращает map, содержащий данные из базы данных.

Исключения

RWEException	Если произошла ошибка при чтении файла.
------------------------------	---

7.7.2.2 writeLog()

```
void rw::writeLog (
    const std::string & path,
    const Exception & e )
```

Функция для записи лога.

Аргументы

path	Путь к файлу логов.
e	Исключение, которое нужно записать в лог.

Исключения

RWEException	Если произошла ошибка при записи файла.
------------------------------	---

7.8 rw.h

[См. документацию.](#)

```
1 #ifndef RW_H
2 #define RW_H
3
4 #include <map>
5 #include <string>
6 #include "exceptions.h" // Подключение нашего файла с исключениями
7
17 namespace rw
18 {
25     std::map<std::string, std::string> readDataBase(const std::string &path);
26
33     void writeLog(const std::string &path, const Exception &e);
34 }
35
36 #endif // RW_H
```

7.9 Файл

/home/student/Документы/busigin/server/source/modules/interface.h

Определение класса интерфейса.

```
#include <map>
#include <string>
#include "exceptions.h"
```

Граф включаемых заголовочных файлов для interface.h: Граф файлов, в которые включается этот файл:

Классы

- class [Interface](#)

Класс для управления интерфейсом программы.

7.9.1 Подробное описание

Определение класса интерфейса.

Этот файл содержит определения классов для обработки параметров командной строки, чтения базы данных паролей и запуска сервера.

Дата

23.11.2024

Версия

1.0

Автор

Бусыгин Г. С.

7.10 interface.h

[См. документацию.](#)

```
1 #ifndef INTERFACE_H
2 #define INTERFACE_H
3
4 #include <map>
5 #include <string>
6 #include "exceptions.h"
7
20 class Interface
21 {
22 public:
29     Interface();
30
34     void parseArgs(int argc, char *argv[]);
35
36     void showHelp() const;
37
42     std::string &getPathToDB();
43
```

```
48     std::string &getPathToLog();
49
54     std::string &getAddress();
55
60     int getPort() const;
61
62 private:
63     std::string pathToDB;
64     std::string pathToLog;
65     std::string address;
66     int port;
67 };
68
69 #endif // INTERFACE_H
```

7.11 Файл /home/student/Документы/busigin/server/source/main.cpp

Главный файл программы.

```
#include "modules/interface.h"
#include "modules/network.h"
#include "modules/rw.h"
#include <iostream>
```

Граф включаемых заголовочных файлов для main.cpp:

Функции

- void `run` (`Interface` &interface, `Network` &network)
Функция для запуска сервера.
- int `main` (int argc, char *argv[])
Главная функция программы.

7.11.1 Подробное описание

Главный файл программы.

Этот файл содержит функцию `main`, которая инициализирует интерфейс и запускает сервер.

Дата

23.11.2024

Версия

1.0

Автор

Бусыгин Г. С.

7.11.2 Функции

7.11.2.1 main()

```
int main (  
    int argc,  
    char * argv[] )
```

Главная функция программы.

Инициализирует объект [Interface](#) и запускает сервер. Обработывает все исключения, возникающие во время выполнения программы.

Аргументы

argc	Количество аргументов командной строки.
argv	Аргументы командной строки.

Возвращает

Код завершения программы. 0 - успешное завершение, 1 - ошибка.

7.11.2.2 run()

```
void run (  
    Interface & interface,  
    Network & network )
```

Функция для запуска сервера.

Аргументы

interface	Объект интерфейса для получения настроек.
network	Объект сети для управления соединениями.

Предметный указатель

/home/student/Документы/busigin/server/source/main.cpp, 16
28
getPort
/home/student/Документы/busigin/server/source/modules/interface.h, 16
24
Network, 19
/home/student/Документы/busigin/server/source/modules/network.h, 16
22, 23
Exception, 14
/home/student/Документы/busigin/server/source/modules/interface.h, 27
Interface, 15
/home/student/Документы/busigin/server/source/modules/network.h, 21, 22
getAddress, 16
getPathToDB, 16
/home/student/Документы/busigin/server/source/modules/rw.h, 25, 26
getPathToLog, 16
getPort, 16
Interface, 15
isCritical
Exception, 14
main
main.cpp, 28
main.cpp
main, 28
run, 30
Network, 17
auth, 18
calc, 18
getAddress, 19
getDatabase, 19
getPort, 19
Network, 18
wait, 19
NetworkException, 20
readDataBase
rw.h, 25
run
main.cpp, 30
rw.h
readDataBase, 25
writeLog, 26
RWException, 20
wait
Network, 19
what
Exception, 14
writeLog
rw.h, 26
/home/student/Документы/busigin/server/source/main.cpp, 16
28
getPort
/home/student/Документы/busigin/server/source/modules/interface.h, 16
24
Network, 19
/home/student/Документы/busigin/server/source/modules/network.h, 22, 23
Exception, 14
/home/student/Документы/busigin/server/source/modules/interface.h, 27
Interface, 15
/home/student/Документы/busigin/server/source/modules/network.h, 21, 22
getAddress, 16
getPathToDB, 16
/home/student/Документы/busigin/server/source/modules/rw.h, 25, 26
getPathToLog, 16
getPort, 16
Interface, 15
isCritical
Exception, 14
main
main.cpp, 28
main.cpp
main, 28
run, 30
Network, 17
auth, 18
calc, 18
getAddress, 19
getDatabase, 19
getPort, 19
Network, 18
wait, 19
NetworkException, 20
readDataBase
rw.h, 25
run
main.cpp, 30
rw.h
readDataBase, 25
writeLog, 26
RWException, 20
wait
Network, 19
what
Exception, 14
writeLog
rw.h, 26
ArgsDecodeException, 11
auth
Network, 18
AuthException, 11
calc
Network, 18
calculateHash
crutils, 9
crutils, 9
calculateHash, 9
DataDecodeException, 12
Exception, 12
Exception, 13
getFunc, 13
getMessage, 14
getName, 14
getTime, 14
isCritical, 14
what, 14
getAddress
Interface, 16
Network, 19
getDatabase
Network, 19
getFunc
Exception, 13
getMessage
Exception, 14
getName
Exception, 14
getPathToDB
Interface, 16
getPathToLog