

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
5.1.2.2	<code>generateSalt()</code>	10
6	Классы	11
6.1	Класс <code>ArgsDecodeException</code>	11
6.2	Класс <code>AuthException</code>	11
6.2.1	Подробное описание	11
6.2.2	Конструктор(ы)	11
6.2.2.1	<code>AuthException()</code>	11
6.3	Класс <code>DataDecodeException</code>	12
6.3.1	Подробное описание	12
6.3.2	Конструктор(ы)	12
6.3.2.1	<code>DataDecodeException()</code>	12
6.4	Класс <code>Exception</code>	13
6.4.1	Подробное описание	13
6.4.2	Конструктор(ы)	13
6.4.2.1	<code>Exception()</code>	13
6.4.3	Методы	14
6.4.3.1	<code>what()</code>	14
6.5	Класс <code>Interface</code>	14
6.5.1	Подробное описание	15
6.5.2	Конструктор(ы)	15
6.5.2.1	<code>Interface()</code>	15
6.5.3	Методы	15
6.5.3.1	<code>getDbPath()</code>	15
6.5.3.2	<code>getLogPath()</code>	15
6.5.3.3	<code>getNetwork()</code>	16
6.5.3.4	<code>parseArgs()</code>	16
6.5.3.5	<code>readDB()</code>	16

6.5.3.6 run()	16
6.6 Класс IOException	17
6.6.1 Подробное описание	17
6.6.2 Конструктор(ы)	17
6.6.2.1 IOException()	17
6.7 Класс Network	18
6.7.1 Подробное описание	18
6.7.2 Конструктор(ы)	18
6.7.2.1 Network()	19
6.7.3 Методы	19
6.7.3.1 auth()	19
6.7.3.2 calc()	19
6.7.3.3 getAddress()	20
6.7.3.4 getDatabase()	20
6.7.3.5 getPort()	20
6.7.3.6 wait()	20
6.8 Класс NetworkException	21
6.8.1 Подробное описание	21
6.8.2 Конструктор(ы)	21
6.8.2.1 NetworkException()	21
7 Файлы	23
7.1 Файл /home/student/Документы/mamelin/server/source/modules/network.h	23
7.1.1 Подробное описание	23
7.2 network.h	24
7.3 Файл /home/student/Документы/mamelin/server/source/modules/exceptions.h	24
7.3.1 Подробное описание	25
7.4 exceptions.h	25
7.5 Файл /home/student/Документы/mamelin/server/source/modules/crutils.h	26
7.5.1 Подробное описание	26
7.6 crutils.h	27
7.7 Файл /home/student/Документы/mamelin/server/source/modules/interface.h	27
7.7.1 Подробное описание	27
7.8 interface.h	28
7.9 Файл /home/student/Документы/mamelin/server/source/main.cpp	28
7.9.1 Подробное описание	28
7.9.2 Функции	29
7.9.2.1 main()	29
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	13
ArgsDecodeException	11
AuthException	11
DataDecodeException	12
IOException	17
NetworkException	21
Interface	14
Network	18

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	13
Interface	
Класс для управления интерфейсом программы	14
IOException	
Класс для исключений ввода-вывода	17
Network	
Класс для управления сетевым подключением и взаимодействием	18
NetworkException	
Класс для сетевых исключений	21

Глава 4

Список файлов

4.1 Файлы

Полный список документированных файлов.

/home/student/Документы/mamelin/server/source/modules/ crutils.h	
Определения вспомогательных функций для криптографических операций	26
/home/student/Документы/mamelin/server/source/modules/ exceptions.h	
Определение классов исключений	24
/home/student/Документы/mamelin/server/source/modules/ interface.h	
Определение класса интерфейса	27
/home/student/Документы/mamelin/server/source/modules/ network.h	
Определения классов для управления сетевым взаимодействием	23
/home/student/Документы/mamelin/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 generateSalt ()`
Функция для генерации соли.
- `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.

5.1.2.2 generateSalt()

`std::string crutils::generateSalt ()`

Функция для генерации соли.

Возвращает

Соль 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, const std::string &log_path, bool critical=false)

Конструктор класса [AuthException](#).

6.2.1 Подробное описание

Класс для исключений аутентификации.

6.2.2 Конструктор(ы)

6.2.2.1 AuthException()

```
AuthException::AuthException (
    const std::string & message,
    const std::string & func,
    const std::string & log_path,
    bool critical = false )
```

Конструктор класса [AuthException](#).

Аргументы

message	Сообщение об ошибке.
func	Имя функции, в которой возникло исключение.
log_path	Путь к файлу журнала.
critical	Флаг критичности исключения.

Объявления и описания членов классов находятся в файлах:

- `/home/student/Документы/mamelin/server/source/modules/exceptions.h`
- `/home/student/Документы/mamelin/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, const std::string &log_path, bool critical=false)
Конструктор класса [DataDecodeException](#).

6.3.1 Подробное описание

Класс для исключений при декодировании данных.

6.3.2 Конструктор(ы)

6.3.2.1 DataDecodeException()

```
DataDecodeException::DataDecodeException (
    const std::string & message,
    const std::string & func,
    const std::string & log_path,
    bool critical = false )
```

Конструктор класса [DataDecodeException](#).

Аргументы

message	Сообщение об ошибке.
func	Имя функции, в которой возникло исключение.
log_path	Путь к файлу журнала.
critical	Флаг критичности исключения.

Объявления и описания членов классов находятся в файлах:

- /home/student/Документы/mamelin/server/source/modules/[exceptions.h](#)
- /home/student/Документы/mamelin/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 &name, const std::string &message, const std::string &func, const std::string &log_path, bool critical=false)
Конструктор класса [Exception](#).
- const char * [what](#) () const noexcept override
Метод для получения сообщения об ошибке.
- void logException () const
Метод для логирования исключения.

6.4.1 Подробное описание

Базовый класс для исключений.

6.4.2 Конструктор(ы)

6.4.2.1 Exception()

```
Exception::Exception (
    const std::string & name,
    const std::string & message,
    const std::string & func,
    const std::string & log_path,
    bool critical = false )
```

Конструктор класса [Exception](#).

Аргументы

name	Имя исключения.
message	Сообщение об ошибке.
func	Имя функции, в которой возникло исключение.
log_path	Путь к файлу журнала.
critical	Флаг критичности исключения.

6.4.3 Методы

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

6.5 Класс Interface

Класс для управления интерфейсом программы.

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

Открытые члены

- `Interface (int argc, char *argv[])`
Конструктор принимает параметры командной строки.
- `void parseArgs (int argc, char *argv[])`
Метод для парсинга аргументов командной строки.
- `void readDB ()`
Метод для чтения базы данных паролей и записи в атрибут map.
- `void run ()`
Метод для запуска сервера.
- `void showHelp () const`
Метод для вывода справки.
- `Network * getNetwork ()`
Метод для получения объекта `Network`.
- `std::string & getDbPath ()`
Метод для получения пути к файлу базы данных.
- `std::string & getLogPath ()`
Метод для получения пути к файлу логов.

6.5.1 Подробное описание

Класс для управления интерфейсом программы.

6.5.2 Конструктор(ы)

6.5.2.1 Interface()

```
Interface::Interface (
    int argc,
    char * argv[] )
```

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 getDbPath()

```
std::string & Interface::getDbPath ( )
```

Метод для получения пути к файлу базы данных.

Возвращает

Путь к файлу базы данных.

6.5.3.2 getLogPath()

```
std::string & Interface::getLogPath ( )
```

Метод для получения пути к файлу логов.

Возвращает

Путь к файлу логов.

6.5.3.3 getNetwork()

`Network * Interface::getNetwork ()`

Метод для получения объекта [Network](#).

Возвращает

Указатель на объект класса [Network](#).

6.5.3.4 parseArgs()

```
void Interface::parseArgs (
    int argc,
    char * argv[] )
```

Метод для парсинга аргументов командной строки.

Аргументы

argc	Количество аргументов командной строки.
argv	Аргументы командной строки.

Исключения

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

6.5.3.5 readDB()

```
void Interface::readDB ( )
```

Метод для чтения базы данных паролей и записи в атрибут `map`.

Исключения

IOException	Если не удалось открыть или прочитать файл базы данных.
DataDecodeException	Если строки в базе данных имеют некорректный формат или произошла ошибка при чтении файла.

6.5.3.6 run()

```
void Interface::run ( )
```

Метод для запуска сервера.

Исключения

NetworkException	Если возникли ошибки при запуске сервера.
----------------------------------	---

Объявления и описания членов классов находятся в файлах:

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

6.6 Класс IOException

Класс для исключений ввода-вывода.

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

Граф наследования:IOException:

Граф связей класса IOException:

Открытые члены

- [IOException](#) (const std::string &message, const std::string &func, const std::string &log_path, bool critical=false)
Конструктор класса [IOException](#).

6.6.1 Подробное описание

Класс для исключений ввода-вывода.

6.6.2 Конструктор(ы)

6.6.2.1 IOException()

```
IOException::IOException (  
    const std::string & message,  
    const std::string & func,  
    const std::string & log_path,  
    bool critical = false )
```

Конструктор класса [IOException](#).

Аргументы

message	Сообщение об ошибке.
func	Имя функции, в которой возникло исключение.
log_path	Путь к файлу журнала.
critical	Флаг критичности исключения.

Объявления и описания членов классов находятся в файлах:

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

6.7 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, const std::string &log_path)
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.

6.7.1 Подробное описание

Класс для управления сетевым подключением и взаимодействием.

6.7.2 Конструктор(ы)

6.7.2.1 Network()

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

Конструктор класса [Network](#).

Аргументы

address	Адрес сервера.
port	Порт сервера.
database	База данных пользователей.
log_path	Путь к файлу логов.

6.7.3 Методы

6.7.3.1 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.
----------------------------------	---

6.7.3.2 calc()

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

Метод для вычисления суммы значений в векторах.

Исключения

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

6.7.3.3 getAddress()

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

Метод для получения адреса сервера.

Возвращает

Адрес сервера.

6.7.3.4 getDatabase()

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

Метод для получения базы данных пользователей.

Возвращает

База данных пользователей.

6.7.3.5 getPort()

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

Метод для получения порта сервера.

Возвращает

Порт сервера.

6.7.3.6 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/mamelin/server/source/modules/network.h](#)
- [/home/student/Документы/mamelin/server/source/modules/network.cpp](#)

6.8 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, const std::string &log_↵ path, bool critical=false)
Конструктор класса [NetworkException](#).

6.8.1 Подробное описание

Класс для сетевых исключений.

6.8.2 Конструктор(ы)

6.8.2.1 NetworkException()

```
NetworkException::NetworkException (
    const std::string & message,
    const std::string & func,
    const std::string & log_path,
    bool critical = false )
```

Конструктор класса [NetworkException](#).

Аргументы

message	Сообщение об ошибке.
func	Имя функции, в которой возникло исключение.
log_path	Путь к файлу журнала.
critical	Флаг критичности исключения.

Объявления и описания членов классов находятся в файлах:

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

Глава 7

Файлы

7.1 Файл

/home/student/Документы/mamelin/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 @authorsa Мамелин Д. А.

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         const std::string &log_path);
17
18     std::string &getAddress();
19
20     uint16_t &getPort();
21
22     std::map<std::string, std::string> &getDatabase();
23
24     void wait();
25
26     void auth();
27
28     void calc();
29
30     void close();
31
32     void quit();
33 private:
34     std::string address;
35     uint16_t port;
36     std::map<std::string, std::string> database;
37     int socket;
38     int client_socket;
39     std::string log_path;
40 };
41 #endif // NETWORK_H

```

7.3 Файл

/home/student/Документы/mamelin/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 [IOException](#)
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 @authorsa Мамелин Д. А.

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:
29     Exception(const std::string &name,
30               const std::string &message,
31               const std::string &func,
32               const std::string &log_path,
33               bool critical = false);
34
39     const char* what() const noexcept override;
40
44     void logException() const;
45
46 private:
47     std::string name;
48     std::string func;
49     std::string log_path;
50     bool critical;
51     mutable std::string message;
52 };
53
57 class IOException : public Exception {
58 public:
66     IOException(const std::string &message,
67                 const std::string &func,
68                 const std::string &log_path,
69                 bool critical = false);
70 };
71
75 class ArgsDecodeException : public Exception {
76 public:
84     ArgsDecodeException(const std::string &message,
85                          const std::string &func,
86                          const std::string &log_path,
87                          bool critical = false);
88 };
89
93 class DataDecodeException : public Exception {
```

```

94 public:
102     DataDecodeException(const std::string &message,
103                          const std::string &func,
104                          const std::string &log_path,
105                          bool critical = false);
106 };
107
111 class AuthException : public Exception {
112 public:
120     AuthException(const std::string &message,
121                  const std::string &func,
122                  const std::string &log_path,
123                  bool critical = false);
124 };
125
129 class NetworkException : public Exception {
130 public:
138     NetworkException(const std::string &message,
139                      const std::string &func,
140                      const std::string &log_path,
141                      bool critical = false);
142 };
143
144 #endif // EXCEPTIONS_H

```

7.5 Файл

/home/student/Документы/mamelin/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::generateSalt ()`
Функция для генерации соли.
- `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 @authorsa Мамелин Д. А.

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 generateSalt();
8
9     std::string calculateHash(const std::string &data);
10 }
11
12 #endif // CRUTILS_H
```

7.7 Файл

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

Определение класса интерфейса.

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

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

Классы

- class [Interface](#)

Класс для управления интерфейсом программы.

7.7.1 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 @authorsa Мамелин Д. А.

7.8 interface.h

См. документацию.

```
1 #ifndef INTERFACE_H
2 #define INTERFACE_H
3
4 #include <map>
5 #include <string>
6 #include "exceptions.h"
7 #include "network.h"
8
21 class Interface
22 {
23 public:
30     Interface(int argc, char *argv[]);
31
38     void parseArgs(int argc, char *argv[]);
39
45     void readDB();
46
51     void run();
52
56     void showHelp() const;
57
62     Network *getNetwork();
63
68     std::string &getDbPath();
69
74     std::string &getLogPath();
75
76 private:
77     Network *network;
78     std::string db_path;
79     std::string log_path;
80     int port;
81     std::string address;
82     std::map<std::string, std::string> database;
83 };
84
85 #endif // INTERFACE_H
```

7.9 Файл /home/student/Документы/mamelin/server/source/main.cpp

Главный файл программы.

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

Граф включаемых заголовочных файлов для main.cpp:

Функции

- int `main` (int argc, char *argv[])
Главная функция программы.

7.9.1 Подробное описание

Главный файл программы.

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

Дата

23.11.2024

Версия

1.0 @authorsa Мамелин Д. А.

7.9.2 Функции

7.9.2.1 main()

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

Главная функция программы.

Инициализирует объект [Interface](#) и запускает его. 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.

Предметный указатель

[/home/student/Документы/mamelin/server/source/main.cpp](#), 28
[/home/student/Документы/mamelin/server/source/modules/args.h](#), 26, 27
[/home/student/Документы/mamelin/server/source/modules/exceptions.h](#), 24, 25
[/home/student/Документы/mamelin/server/source/modules/network.h](#), 27, 28
[/home/student/Документы/mamelin/server/source/modules/main.cpp](#), 23, 24

ArgsDecodeException, 11
auth
 Network, 19
AuthException, 11
 AuthException, 11
calc
 Network, 19
calculateHash
 crutils, 9
crutils, 9
 calculateHash, 9
 generateSalt, 10
DataDecodeException, 12
 DataDecodeException, 12
Exception, 13
 Exception, 13
 what, 14
generateSalt
 crutils, 10
getAddress
 Network, 19
getDatabase
 Network, 20
getDbPath
 Interface, 15
getLogPath
 Interface, 15
getNetwork
 Interface, 15
getPort
 Network, 20
Interface, 14
 getDbPath, 15
 getLogPath, 15
main
 Network, 15
 Interface, 15
 parseArgs, 16
 readDB, 16
 modules/exceptions.h, 17
 IOException, 17
 IOException, 17
 modules/network.h, 17
 main.cpp, 29
 main.cpp
 main, 29
Network, 18
 auth, 19
 calc, 19
 getAddress, 19
 getDatabase, 20
 getPort, 20
 Network, 18
 wait, 20
NetworkException, 21
 NetworkException, 21
parseArgs
 Interface, 16
readDB
 Interface, 16
run
 Interface, 16
wait
 Network, 20
what
 Exception, 14