

Программная реализация сетевого сервера

1

Создано системой Doxygen 1.9.1

1 Алфавитный указатель классов	1
1.1 Классы	1
2 Список файлов	3
2.1 Файлы	3
3 Классы	5
3.1 Класс Auth	5
3.1.1 Методы	5
3.1.1.1 authenticate()	5
3.1.1.2 hashPassword()	6
3.2 Класс ErrorTracker	6
3.2.1 Конструктор(ы)	6
3.2.1.1 ErrorTracker()	6
3.2.2 Методы	7
3.2.2.1 writeToLog()	7
3.3 Класс Server	7
3.3.1 Конструктор(ы)	7
3.3.1.1 Server()	7
3.3.2 Методы	8
3.3.2.1 run()	8
4 Файлы	9
4.1 Файл auth.cpp	9
4.1.1 Подробное описание	9
4.2 Файл auth.h	9
4.2.1 Подробное описание	10
4.3 Файл errors.cpp	10
4.3.1 Подробное описание	11
4.4 Файл errors.h	11
4.4.1 Подробное описание	12
4.5 Файл main.cpp	12
4.5.1 Подробное описание	13
4.5.2 Функции	13
4.5.2.1 main()	13
4.6 Файл server.cpp	14
4.6.1 Подробное описание	14
4.7 Файл server.h	14
4.7.1 Подробное описание	15
Предметный указатель	17

Глава 1

Алфавитный указатель классов

1.1 Классы

Классы с их кратким описанием.

Auth	5
ErrorTracker	6
Server	7

Глава 2

Список файлов

2.1 Файлы

Полный список документированных файлов.

auth.cpp	Реализация класса Auth	9
auth.h	Заголовочный файл для модуля AUTH	9
errors.cpp	Реализация класса ErrorTracker	10
errors.h	Заголовочный файл для модуля ERRORS	11
main.cpp	Главный файл программы	12
server.cpp	Реализация класса Server	14
server.h	Заголовочный файл для модуля SERVER	14

Глава 3

Классы

3.1 Класс Auth

Открытые статические члены

- static std::string generateRandomSalt ()
- static std::string [hashPassword](#) (const std::string &password, const std::string &salt)
Хэширование пароля с использованием соли
- static bool [authenticate](#) (const std::string &username, const std::string &password, const std::string &databaseFile)
Аутентификация пользователя

3.1.1 Методы

3.1.1.1 authenticate()

```
bool Auth::authenticate (  
    const std::string & username,  
    const std::string & password,  
    const std::string & databaseFile ) [static]
```

Аутентификация пользователя

Этот метод проверяет подлинность пользователя, сравнивая предоставленные имя пользователя и пароль с данными из базы данных.

Аргументы

username	Имя пользователя
password	Пароль
databaseFile	Путь к файлу базы данных

Возвращает

true, если аутентификация успешна, false в противном случае

3.1.1.2 hashPassword()

```
std::string Auth::hashPassword (
    const std::string & password,
    const std::string & salt ) [static]
```

Хэширование пароля с использованием соли

Этот метод принимает пароль и соль, объединяет их, затем хэширует с использованием алгоритма MD5.

Аргументы

password	Пароль
salt	Соль

Возвращает

Хэшированный пароль в виде строки

Объявления и описания членов классов находятся в файлах:

- [auth.h](#)
- [auth.cpp](#)

3.2 Класс ErrorTracker

Открытые члены

- [ErrorTracker](#) (const std::string &logFilePath)
Конструктор класса [ErrorTracker](#).
- void [writeToLog](#) (const std::string &message, bool isError)
Запись ошибок

3.2.1 Конструктор(ы)

3.2.1.1 ErrorTracker()

```
ErrorTracker::ErrorTracker (
    const std::string & logFileName )
```

Конструктор класса [ErrorTracker](#).

Аргументы

logFileName	Имя файла журнала
-------------	-------------------

3.2.2 Методы

3.2.2.1 writeToLog()

```
void ErrorTracker::writeToLog (
    const std::string & message,
    bool isError )
```

Запись ошибок

Аргументы

message	Сообщение
isError	Значение ошибки

Объявления и описания членов классов находятся в файлах:

- [errors.h](#)
- [errors.cpp](#)

3.3 Класс Server

Открытые члены

- [Server](#) ([ErrorTracker](#) *errorTracker)
Конструктор класса [Server](#).
- void [run](#) (int port, const std::string &databaseFile, const std::string &logFile)
Запуск сервера

3.3.1 Конструктор(ы)

3.3.1.1 Server()

```
Server::Server (
    ErrorTracker * errorTracker )
```

Конструктор класса [Server](#).

Аргументы

errorTracker	Указатель на объект ErrorTracker для отслеживания ошибок
--------------	--

3.3.2 Методы

3.3.2.1 run()

```
void Server::run (  
    int port,  
    const std::string & databaseFile,  
    const std::string & logFile )
```

Запуск сервера

Аргументы

port	Порт сервера
databaseFile	Путь к файлу базы данных
logFile	Путь к файлу журнала

Объявления и описания членов классов находятся в файлах:

- [server.h](#)
- [server.cpp](#)

Глава 4

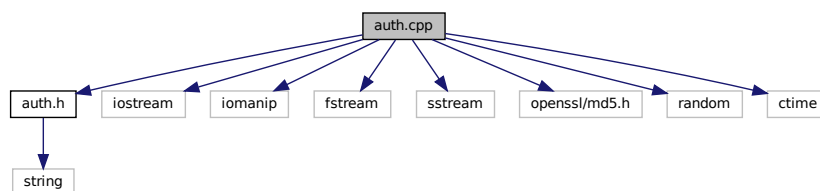
Файлы

4.1 Файл auth.cpp

Реализация класса [Auth](#).

```
#include "auth.h"  
#include <iostream>  
#include <iomanip>  
#include <fstream>  
#include <sstream>  
#include <openssl/md5.h>  
#include <random>  
#include <ctime>
```

Граф включаемых заголовочных файлов для auth.cpp:



4.1.1 Подробное описание

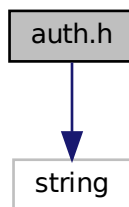
Реализация класса [Auth](#).

4.2 Файл auth.h

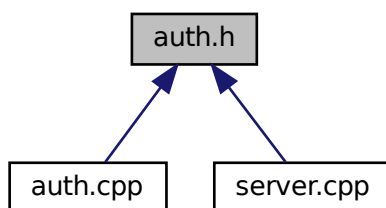
Заголовочный файл для модуля AUTH.

```
#include <string>
```

Граф включаемых заголовочных файлов для auth.h:



Граф файлов, в которые включается этот файл:



Классы

- class [Auth](#)

4.2.1 Подробное описание

Заголовочный файл для модуля AUTH.

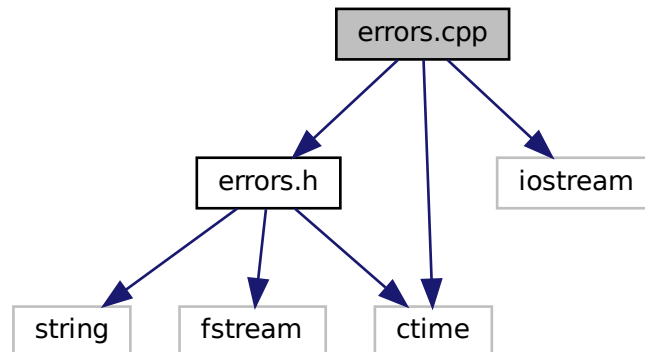
4.3 Файл errors.cpp

Реализация класса [ErrorTracker](#).

```
#include "errors.h"  
#include <iostream>
```

```
#include <ctime>
```

Граф включаемых заголовочных файлов для errors.cpp:



4.3.1 Подробное описание

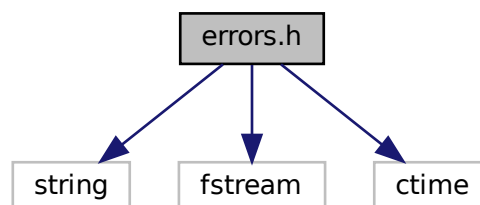
Реализация класса [ErrorTracker](#).

4.4 Файл errors.h

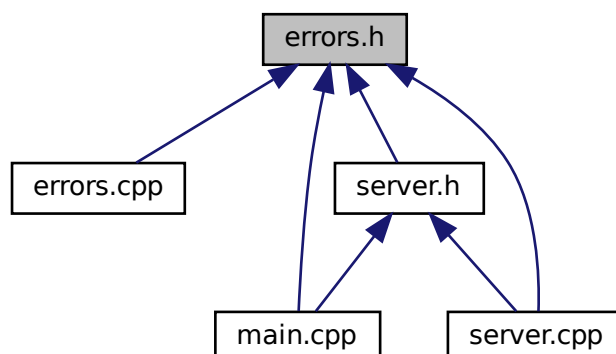
Заголовочный файл для модуля ERRORS.

```
#include <string>
#include <fstream>
#include <ctime>
```

Граф включаемых заголовочных файлов для errors.h:



Граф файлов, в которые включается этот файл:



Классы

- class [ErrorTracker](#)

4.4.1 Подробное описание

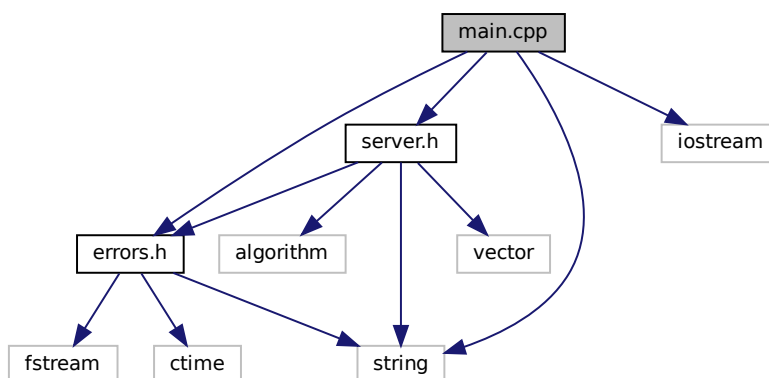
Заголовочный файл для модуля ERRORS.

4.5 Файл main.cpp

Главный файл программы

```
#include "server.h"
#include "errors.h"
#include <iostream>
#include <string>
```

Граф включаемых заголовочных файлов для main.cpp:



Функции

- `int main (int argc, char *argv[])`
Точка входа в программу

4.5.1 Подробное описание

Главный файл программы

Автор

Никитина М. А. 22ПИ2

Версия

1.0

Дата

25.01.2024

Авторство

ИБСТ ПГУ

4.5.2 Функции

4.5.2.1 main()

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

Точка входа в программу

Аргументы

argc	Количество аргументов командной строки
argv	Массив аргументов командной строки

Возвращает

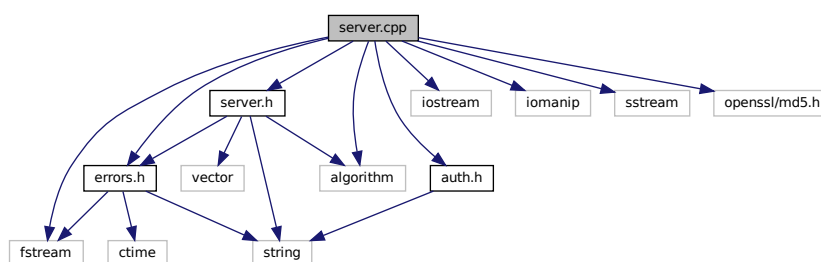
Код завершения программы

4.6 Файл server.cpp

Реализация класса [Server](#).

```
#include "server.h"
#include "auth.h"
#include "errors.h"
#include <iostream>
#include <iomanip>
#include <fstream>
#include <sstream>
#include <openssl/md5.h>
#include <algorithm>
```

Граф включаемых заголовочных файлов для server.cpp:



4.6.1 Подробное описание

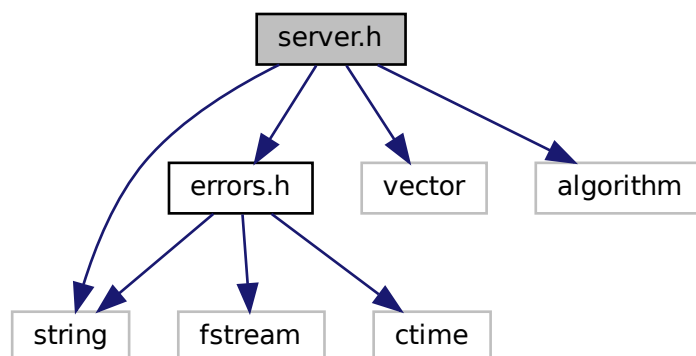
Реализация класса [Server](#).

4.7 Файл server.h

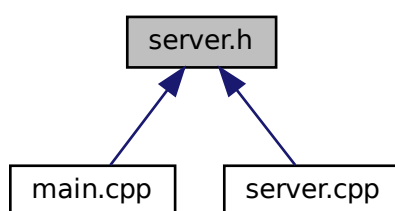
Заголовочный файл для модуля SERVER.

```
#include "errors.h"
#include <string>
#include <vector>
#include <algorithm>
```

Граф включаемых заголовочных файлов для server.h:



Граф файлов, в которые включается этот файл:



Классы

- class [Server](#)

4.7.1 Подробное описание

Заголовочный файл для модуля SERVER.

Предметный указатель

- Auth, [5](#)
 - authenticate, [5](#)
 - hashPassword, [6](#)
- auth.cpp, [9](#)
- auth.h, [9](#)
- authenticate
 - Auth, [5](#)
- errors.cpp, [10](#)
- errors.h, [11](#)
- ErrorTracker, [6](#)
 - ErrorTracker, [6](#)
 - writeToLog, [7](#)
- hashPassword
 - Auth, [6](#)
- main
 - main.cpp, [13](#)
- main.cpp, [12](#)
 - main, [13](#)
- run
 - Server, [8](#)
- Server, [7](#)
 - run, [8](#)
 - Server, [7](#)
- server.cpp, [14](#)
- server.h, [14](#)
- writeToLog
 - ErrorTracker, [7](#)