Calculator Server

1.0

Создано системой Doxygen 1.9.1

Глава 1

Иерархический список классов

1.1 Иерархия классов

Иерархия классов.

Глава 2

Алфавитный указатель классов

2.1 Классы

Классы с их кратким описанием.

Глава 3

Список файлов

3.1 Файлы

Полный список документированных файлов.

Глава 4

Классы

4.1 Класс Auth

Класс аутентификации пользователей

#include <auth.h>

Закрытые члены

- std::string salt_gen ()
  Метод генерации значения SALT.
- std::string pw_hash (std::string salt, std::string password)
  Метод хэширования пароля
- bool id_check (std::string id, std::map< std::string, std::string > base_c)
  Метод проверки наличия в базе пользователей указанного ID.
- void operator() (int sock, Error &er)
  Перегрузка оператора () для класса аутентификации
- void auth (Error &er)
  Метод, проводящий аутентификацию пользователя
- std::string string_recv (Error &er)
  Метод для получения строки от клиента
- std::map< std::string, std::string > base_read (std::string file_name,)
  Метод чтения базы пользователей

Закрытые данные

- int SALT
  значение SALT для хэширования пароля
- int work_sock
  переменная сокета
- std::map< std::string, std::string > base_cont
  ассоциативный массив id - пароль

4.1.1 Подробное описание

Класс аутентификации пользователей

В классе используются функции для сетевого обмена данными. В конструкторе сокету присваивается значение -1, чтобы исключить использование сокета который не был получен от другого класса

4.1.2 Методы

4.1.2.1 auth()

void Auth::auth (
Error & er) [private]

Метод, проводящий аутентификацию пользователя

Выполняет такие методы класса id_check(), salt_gen(), pw_hash() и pw_check(), а также принимает с помощью сетевых функций id и хэшированный пароль от пользователя, сам метод после идентификации отправляет клиету значение SALT.

Аргументы

in	er	объект класса исключений для установки параметра критичности

Предупреждения

Выполняется только с открытым соединением

Исключения

system_error,если	произошла ошибка в сетевом взаимодействии

4.1.2.2 base_read()

std::map< std::string, std::string > Auth::base_read (
std::string file_name) [private]

Метод чтения базы пользователей

Данные из файла с информацией о пользователях читаются в ассоциативный массив "ID-пароль" для дальнейшего использования

Аргументы

in	file_name	имя файла с базой пользователей

Возвращает

Ассоциативный массив "ID-пароль"

Исключения

system_error,если	произошла ошибка открытия файла
length_error,если	файл базы пустой

4.1.2.3 id_check()

```
bool Auth::id_check (
            std::string id,
            std::map< std::string, std::string > base_c )   [private]
```

Метод проверки наличия в базе пользователей указанного ID.

Аргументы

in	id	ID, введенный клиентом
in	base↩ _c	контейнер для хранения базы пользователей

Исключения

auth_error,если	идентификация не удалась

Возвращает

Значение true, если идентификация прошла успешно

4.1.2.4 operator()()

```
void Auth::operator() (
            int sock,
            Error & er )   [private]
```

Перегрузка оператора () для класса аутентификации

Является "сеттером" для атрибута work_sock, а также выполняет метод auth()

Аргументы

in	er	объект класса исключений для установки параметра критичности
in	sock	значение сокета, полученное от модуля соединения

4.1.2.5 pw_hash()

```
std::string Auth::pw_hash (
            std::string salt,
            std::string password )  [private]
```

Метод хэширования пароля

Пароль хэшируется с помощью ранее полученного значения SALT функциями бибилиотеки CryptoPP. Хэшированный пароль используется для авторизации пользователя

Аргументы

in	salt	значение SALT
in	password	пароль для хэшированмя

Возвращает

Хэшированный пароль

4.1.2.6 salt_gen()

```
std::string Auth::salt_gen ( )  [private]
```

Метод генерации значения SALT.

SALT генерируется на основе системного времени с помощью библиотеки CryptoPP. Входных параментров нет

Возвращает

Сгенерированное значение SALT в виде строки

4.1.2.7 string_recv()

```
std::string Auth::string_recv (
            Error & er )  [private]
```

Метод для получения строки от клиента

Чтение строки реализовывается с помощью цикла удвоения буфера, чтобы поддерживать чтение сообщений любой длины. Используется для получения пароля и ID для аутентификации

Аргументы

in	er	объект класса исключений для установки параметра критичности

Возвращает

Строка, полученная, от клиента

Исключения

system_error,если	произошла ошибка приема данных от клиента

Объявления и описания членов классов находятся в файлах:

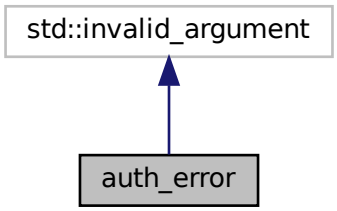
- auth.h
- auth.cpp

4.2 Класс auth_error

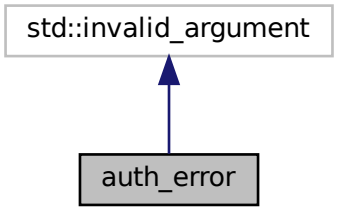
Пользовательский класс исключений

#include <server_error.h>

Граф наследования:auth_error:



Граф связей класса auth_error:



Открытые члены

- auth_error (const std::string &s)
- auth_error (const char *s)

4.2.1 Подробное описание

Пользовательский класс исключений

Класс созан для отдельного определения исключений возникших в при аутентификации пользоваетля

Объявления и описания членов класса находятся в файле:

- server_error.h

4.3 Класс Calculation

Класс вычислений для вектора чисел

#include <calc.h>

Открытые члены

- Calculation ()
  Конструктор класса вычислений
- void calc (Error &er)
  Метод реализации вычислений
- float multip (std::vector< float > vector)
  Метод премножения значений вектора
- void operator() (int sock, Error &er)
  Перегрузка оператора () для класса вычислений
- float overflow (float res)
  Метод проверки на переполнение

Закрытые данные

- uint32_t count

  количество векторов
- uint32_t size

  размер верктора
- int work_sock

  значения сокета

4.3.1 Подробное описание

Класс вычислений для вектора чисел

С помощью метов класса происходит получение данныых о количестве, размере и значений векторов чисел. После этого происходит вычисление произведения для каждого вектора

Предупреждения

Используется только после открытия соединения с клиентом

4.3.2 Конструктор(ы)

4.3.2.1 Calculation()

Calculation::Calculation ( ) [inline]

Конструктор класса вычислений

Присваивает -1 значению сокета, полученного от класса соединения

4.3.3 Методы

4.3.3.1 calc()

void Calculation::calc (
Error & er)

Метод реализации вычислений

Внутри метода с помощью сетевых функций происходит получение количества, размера и значений векторов, используются функции с calc() и overflow(), чтобы получить результат

Аргументы

in	er	объект класса исключений для установки параметра критичности

Предупреждения

Работает только при открытом соединении

4.3.3.2 multip()

```
float Calculation::multip (
            std::vector< float > vector )
```

Метод премножения значений вектора

Аргументы

vector	вектор значений, которые будут перемножаться

Возвращает

Результат переменожения чисел, из которых состоит вектор

Исключения

system_error,если	произолша обшибка получения данных

4.3.3.3 operator()()

```
void Calculation::operator() (
            int sock,
            Error & er )
```

Перегрузка оператора () для класса вычислений

Является "сеттером" для атрибута work_sock, а также выполняет метод calc()

Аргументы

in	er	объект класса исключений для установки параметра критичности
	sock	значение сокета, полученное от модуля соединения

4.3.3.4 overflow()

```
float Calculation::overflow (
                float res )
```

Метод проверки на переполнение

Проверяет на переполнение результат вычислений при переполнении использует переменную из библиотеки "float.h" FLT_MAX(переполнение вниз: -FLT_MAX, вверх: FLT_MAX). При отсутсвиии переполнения возвращает значение из параметра

Аргументы

res	Числи для проверки на переполнение

Возвращает

При переполнении FLT_MAX или -FLT_MAX, без переполнения входной параметр

Объявления и описания членов классов находятся в файлах:

- calc.h
- calc.cpp

4.4 Класс Connection

Класс реализации соединения между клиентом и сервером

#include <connection.h>

Открытые члены

- Connection (unsigned short port)
  Конструктор класса соединения
- ~Connection ()
  Деструктор для класса соединения
- void connect (Calculation &clc, Auth &ath, Error &er)
  Метод для установки соединения

Закрытые данные

- int sock
  переменная сокета
- std::unique_ptr< sockaddr_in > serv_addr
  умный указатель на адресную структуру сервера
- std::unique_ptr< sockaddr_in > client_addr
  умный указатель на адресную структуру клиента
- int queue_len =5
  длина очереди ожидания

4.4.1 Подробное описание

Класс реализации соединения между клиентом и сервером

В классе создается сокет для адреса сервера, производится привязка к этому адресу. После этого сервер переходит в режим ожидания соединения

4.4.2 Конструктор(ы)

4.4.2.1 Connection()

```
Connection::Connection (
            unsigned short port )
```

Конструктор класса соединения

В конструкторе создается сокет для сервера, происходит настройка сокета(возможность повторного использования и время ожидания соединения) и происходит привязка сокета к адресу сервера

Аргументы

port	порт по, которому будет происходить соединение

4.4.2.2 ~Connection()

```
Connection::~Connection ( )  [inline]
```

Деструктор для класса соединения

Внутри деструктора закрывается сокет для сервера

4.4.3 Методы

4.4.3.1 connect()

```
void Connection::connect (
            Calculation & clc,
            Auth & ath,
            Error & er )
```

Метод для установки соединения

Внутри метода с помощью методов сетевого взаимодейтсвия запускается бесконечный цикл ожидания соединения. После установки соединения с помощью перегруженного оператора () данные о сокете от метода accept передаются классам аутентификации и вычислений для реализации дальнейшей работы программы. Метод не возвраащет значений

Аргументы

in	clc	объект класса вычислений
in	ath	объект класса аутентификации
in	er	объект класса исключений для установки параметра критичности

Объявления и описания членов классов находятся в файлах:

- connection.h
- connection.cpp

4.5 Класс Error

Открытые члены

- std::string write_log (std::string file_name, std::string e_str)

  Метод записи сообщений об ошиках в лог
- void critic_set ()

  метод установки параметра критичности для ошибки

Закрытые данные

- bool iscritical =false

  метка, чтобы показать критичность ошибки

4.5.1 Методы

4.5.1.1 write_log()

```
std::string Error::write_log (
                std::string file_name,
                std::string e_str )
```

Метод записи сообщений об ошиках в лог

Сообщение об исключении выводится в формате: "текущее системное время - текст исключения"

Аргументы

file_name	название лог-файла, в который будут записываться сообщения об ошибках, в случае его отсутсвия он будет создан
e_str	строка с описанием исключения, которая была задана в кострукторе исключения в другом модуле

Возвращает

Строка формата "текущее системное время - текст исключения"

Объявления и описания членов классов находятся в файлах:

- server_error.h
- server_error.cpp

4.6 Класс Interface

Класс интерфейса взаимодействия с пользователем

#include <interface.h>

Открытые члены

- Interface ()
  Конструктор в котором обнуляются все значения атрибутов класса
- bool set_options (int argscount, char *argvectors[], Error &er)
  Метод установки опций интерфейса
- std::string get_basefile () const
  "Геттер" для атрибута имени файла с базой пользователей
- uint32_t get_port () const
  "Геттер" для атрибута порта
- std::string get_logfile () const
  "Геттер" для атрибута имени лог-файла

Закрытые данные

- uint32_t port
  порт для соединения
- std::string base_file
  название файла, содержащего базу пользователей
- std::string log_file
  название файла, в который будут записываться сообщения об ошибках

4.6.1 Подробное описание

Класс интерфейса взаимодействия с пользователем

Конструтор обнуляет все значения. Интрефейс реализован для командной строки с помощью библиотеки boost_program_options

Предупреждения

Использование несуществующих опций может привести к аварийному останову

4.6.2 Конструктор(ы)

4.6.2.1 Interface()

Interface::Interface ( ) [inline]

Конструктор в котором обнуляются все значения атрибутов класса

Поддерживается создание конструктора без параметров. Внутри коструктора вызов методов не происходит

4.6.3 Методы

4.6.3.1 get_basefile()

std::string Interface::get_basefile ( ) const

"Геттер" для атрибута имени файла с базой пользователей

Используется для передачи имени файла с базой пользователей модулю аутентификации для чтения этой базы, т.к. атрибут приватный

Возвращает

Имя файла, содержащего данные о пользователях

4.6.3.2 get_logfile()

std::string Interface::get_logfile ( ) const

"Геттер" для атрибута имени лог-файла

Используется для передачи имени лог-файла функции записи сообщений об ошиках, т.к атрибут приватный

Возвращает

Имя лог-файла

4.6.3.3 get_port()

```
uint32_t Interface::get_port ( ) const
```

"Геттер" для атрибута порта

Используется для передачи порта классу сетевого взаимодействия для реализации сетевых функций, т.к. атрибут приватный

Возвращает

Порт, по которому будет происходить соединение

4.6.3.4 set_options()

```
bool Interface::set_options (
            int argscount,
            char * argvectors[],
            Error & er )
```

Метод установки опций интерфейса

Реализация с помощью boost_program_options. Поддержка опций: "порт", "файл базы пользователей" и "файл для записи ошибок". Метод значений не возвращает

Аргументы

in	argscount	количество агрументов командной строки, принимается из параметров функции main
in	argvectors	указатель на массив со значениями агрументов командной строки, также принимается из параметров функции main

Возвращает

False или true, означающее, нужно ля запускать программу(для вызова справки)

Предупреждения

Использование несуществующих опций может привести к аварийному останову

Аргументы

in	er	объект класса исключений для установки параметра критичности

Объявления и описания членов классов находятся в файлах:

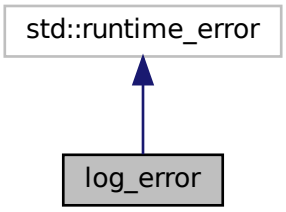
- interface.h
- interface.cpp

4.7 Класс log_error

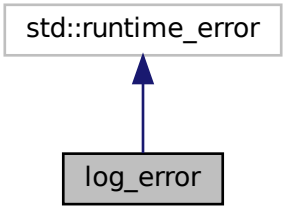
Пользовательский класс исключений в методе записи лога

#include <server_error.h>

Граф наследования:log_error:



Граф связей класса log_error:



Открытые члены

- log_error (const std::string &s)
- log_error (const char *s)

4.7.1 Подробное описание

Пользовательский класс исключений в методе записи лога

Создан для того чтобы специально отслеживать исключения, которые возникают при записи в файл лога, т.к. в общем случае сообщение после обработки записывается в файл логов, в этом случае это, очевидно, невозможно

Объявления и описания членов класса находятся в файле:

- server_error.h

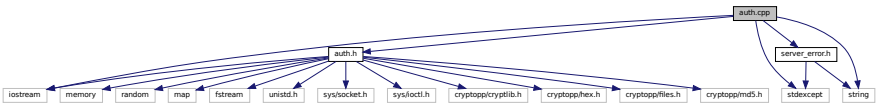
Глава 5

Файлы

5.1 Файл auth.cpp

CPP-файл модуля аутентификации курсовой работы

#include "auth.h"
#include "server_error.h"
#include <iostream>
#include <string>
#include <stdexcept>
Граф включаемых заголовочных файлов для auth.cpp:



5.1.1 Подробное описание

CPP-файл модуля аутентификации курсовой работы

Автор

Гастиев Артур

Версия

1.0

Дата

16.12.2023

Предупреждения

Студенческая работа

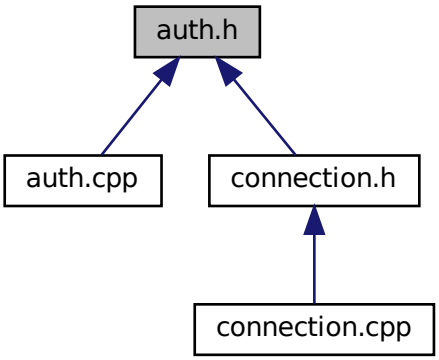
5.2 Файл auth.h

Заголовочный файл для модуля аутентификации курсовой работы

```
#include <iostream>
#include <memory>
#include <random>
#include <map>
#include <fstream>
#include <unistd.h>
#include <sys/socket.h>
#include <sys/ioctl.h>
#include <cryptopp/cryptlib.h>
#include <cryptopp/hex.h>
#include <cryptopp/files.h>
#include <cryptopp/md5.h>
```

Граф включаемых заголовочных файлов для auth.h:

Граф файлов, в которые включается этот файл:



Классы

- class Auth

  Класс аутентификации пользователей

Макросы

- #define CRYPTOPP_ENABLE_NAMESPACE_WEAK 1

5.2.1 Подробное описание

Заголовочный файл для модуля аутентификации курсовой работы

Автор

Гастиев Артур

Версия

1.0

Дата

16.12.2023

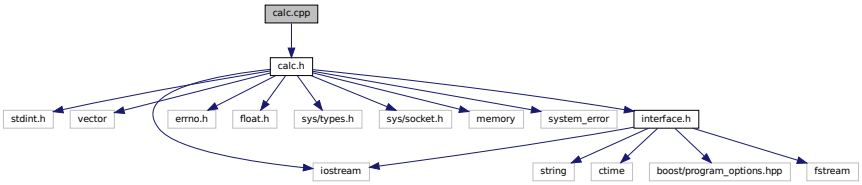
Предупреждения

Студенческая работа

5.3 Файл calc.cpp

CPP-файл модуля вычислений курсовой работы

#include "calc.h"
Граф включаемых заголовочных файлов для calc.cpp:



5.3.1 Подробное описание

CPP-файл модуля вычислений курсовой работы

Автор

Гастиев Артур

Версия

1.0

Дата

16.12.2023

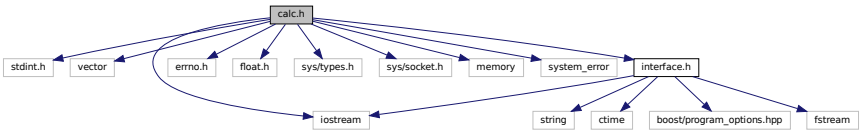
Предупреждения

Студенческая работа

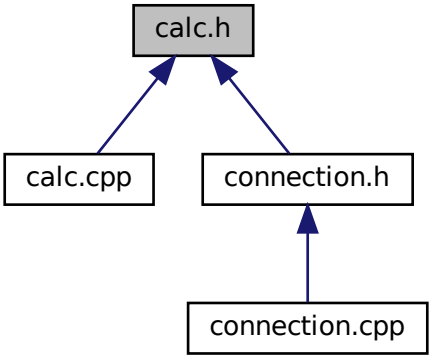
5.4 Файл calc.h

Заголовочный файл для модуля вычислений курсовой работы

#include <stdint.h>
#include <vector>
#include <iostream>
#include <errno.h>
#include <float.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <memory>
#include <system_error>
#include "interface.h"
Граф включаемых заголовочных файлов для calc.h:



Граф файлов, в которые включается этот файл:



Классы

- class Calculation

  Класс вычислений для вектора чисел

5.4.1 Подробное описание

Заголовочный файл для модуля вычислений курсовой работы

Автор

Гастиев Артур

Версия

1.0

Дата

16.12.2023

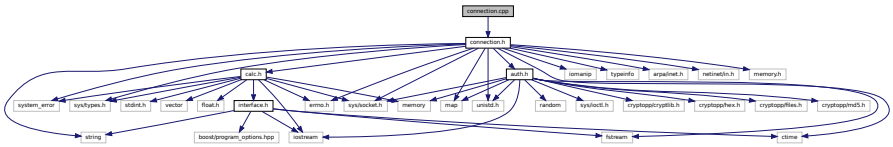
Предупреждения

Студенческая работа

5.5 Файл connection.cpp

CPP-файл модуля соединения курсовой работы

#include "connection.h"
Граф включаемых заголовочных файлов для connection.cpp:



5.5.1 Подробное описание

CPP-файл модуля соединения курсовой работы

Автор

Гастиев Артур

Версия

1.0

Дата

16.12.2023

Предупреждения

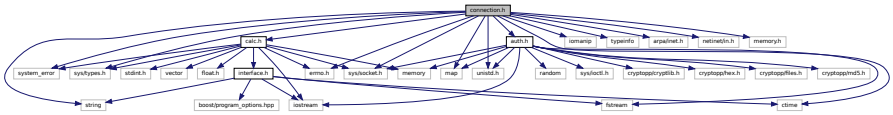
Студенческая работа

5.6 Файл connection.h

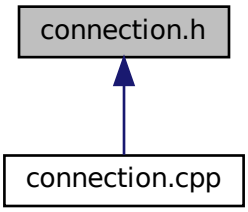
Заголовочный файл для модуля моединения курсовой работы

```
#include <string>
#include <iomanip>
#include <map>
#include <typeinfo>
#include <system_error>
#include <unistd.h>
#include <arpa/inet.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <errno.h>
#include <memory.h>
#include "calc.h"
#include "auth.h"
#include <ctime>
```

Граф включаемых заголовочных файлов для connection.h:



Граф файлов, в которые включается этот файл:



Классы

- class Connection

  Класс реализации соединения между клиентом и сервером

5.6.1 Подробное описание

Заголовочный файл для модуля моединения курсовой работы

Автор

Гастиев Артур

Версия

1.0

Дата

16.12.2023

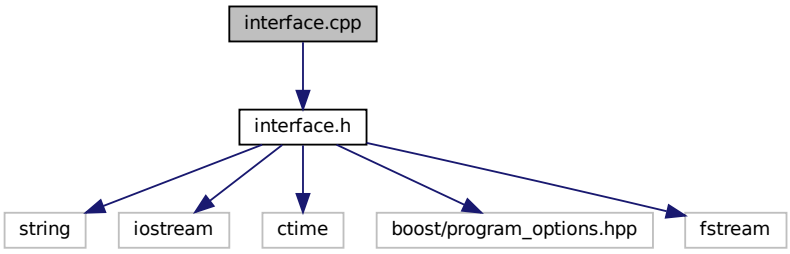
Предупреждения

Студенческая работа

5.7 Файл interface.cpp

CPP-файл для модуля интерфейса курсовой работы

#include "interface.h"
Граф включаемых заголовочных файлов для interface.cpp:



Переменные

- struct {
  uint32_t p =33333
  std::string b ="base.txt"
  std::string l ="log.txt"
  } params

5.7.1 Подробное описание

CPP-файл для модуля интерфейса курсовой работы

Автор

Гастиев Артур

Версия

1.0

Дата

16.12.2023

Предупреждения

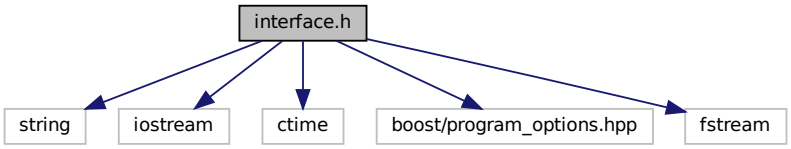
Студенческая работа

5.8 Файл interface.h

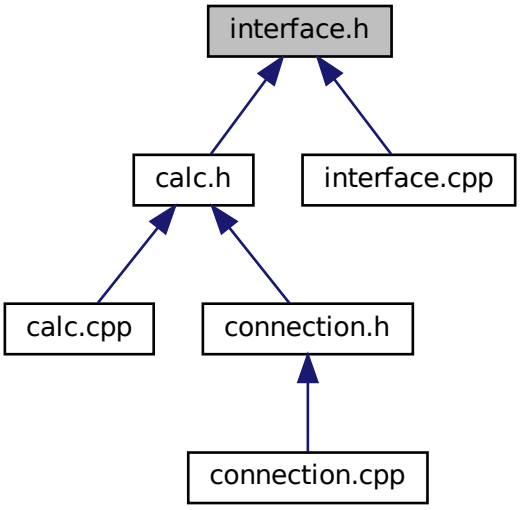
Заголовочный файл для модуля интерфейса курсовой работы

```
#include <string>
#include <iostream>
#include <ctime>
#include <boost/program_options.hpp>
#include <fstream>
```

Граф включаемых заголовочных файлов для interface.h:



Граф файлов, в которые включается этот файл:



Классы

- class Interface

  Класс интерфейса взаимодействия с пользователем

5.8.1 Подробное описание

Заголовочный файл для модуля интерфейса курсовой работы

Автор

Гастиев Артур

Версия

1.0

Дата

16.12.2023

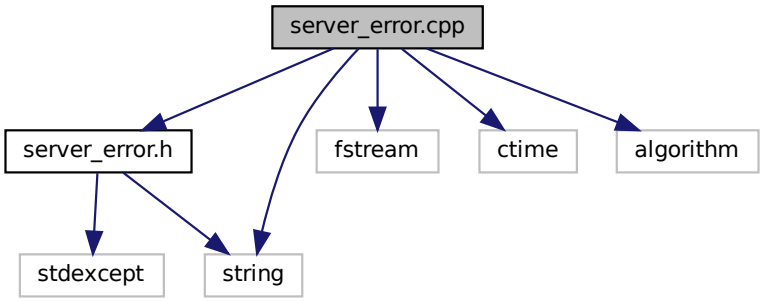
Предупреждения

Студенческая работа

5.9 Файл server_error.cpp

CPP для для модуля исключений

#include "server_error.h"
#include <string>
#include <fstream>
#include <ctime>
#include <algorithm>
Граф включаемых заголовочных файлов для server_error.cpp:



5.9.1 Подробное описание

CPP для для модуля исключений

Автор

Гастиев Артур

Версия

1.0

Дата

22.12.2023

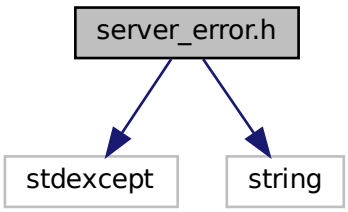
Предупреждения

Студенческая работа

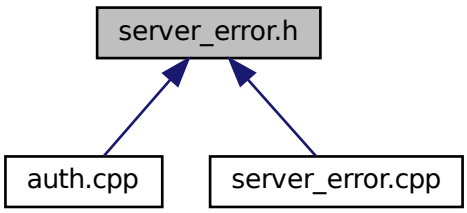
5.10 Файл server_error.h

Заголовочный файл для для модуля исключений и пользовательских классов исключений

#include <stdexcept>
#include <string>
Граф включаемых заголовочных файлов для server_error.h:



Граф файлов, в которые включается этот файл:



Классы

- class log_error

  Пользовательский класс исключений в методе записи лога
- class auth_error

  Пользовательский класс исключений
- class Error

5.10.1 Подробное описание

Заголовочный файл для для модуля исключений и пользовательских классов исключений

Автор

Гастиев Артур

Версия

1.0

Дата

22.12.2023

Предупреждения

Студенческая работа

Пользовательские классы исключений используются для того, чтобы различить какая именно и в каком мудуле произошла ошибка

Предметный указатель