Разработка клиент-серверных приложений с подсистемой аутентификации

Создано системой Doxygen 1.9.4

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 base

Класс для чтения базы данных

#include <base.h>

Открытые члены

- void connect (std::string f)
  Установка соединения с базой данных
- std::map< std::string, std::string > get_data ()
  Получить базу данных
- bool has_login (const std::string &login) const
  Проверка наличия логина в базе данных
- std::string get_password (const std::string &login) const
  Получение пароля по логину

Закрытые данные

- std::map< std::string, std::string > data_base
  Контейнер "логин+пароль".

4.1.1 Подробное описание

Класс для чтения базы данных

Контейнер data_base хранит в себе логин и пароль пользователя. Для получения базы используется метод get_data().

4.1.2 Методы

4.1.2.1 connect()

```
void base::connect (
            std::string f )
```

Установка соединения с базой данных

Читает из файла строку базы данных.

Аргументы

in	f	Путь к файлу базы данных.

Исключения

crit_err	Если файл не открывается, либо несоответствие формату строки "логин:пароль".

4.1.2.2 get_data()

std::map< std::string, std::string > base::get_data ( ) [inline]

Получить базу данных

Возвращает

Контейнер с базой данных.

4.1.2.3 get_password()

std::string base::get_password (
const std::string & login) const [inline]

Получение пароля по логину

Аргументы

in	login	Логин.

Возвращает

Пароль, если логин существует, иначе пустая строка.

4.1.2.4 has_login()

bool base::has_login (
const std::string & login) const [inline]

Проверка наличия логина в базе данных

Аргументы

in	login	Логин для проверки.

Возвращает

true, если логин существует, иначе false.

Объявления и описания членов классов находятся в файлах:

- base.h
- base.cpp

4.2 Класс calc

Класс для вычисления суммы элементов вектора

#include <calc.h>

Открытые члены

- calc (std::vector< uint32_t > chisla)
  Конструктор для вычисления суммы элементов вектора
- uint32_t send_res ()
  Метод для отправки результата

Открытые атрибуты

- uint32_t res
  Переменная, в которую будет записан результат

4.2.1 Подробное описание

Класс для вычисления суммы элементов вектора

Вектор указывается в параметрах конструктора. Для получения результата вычислений используется метод send_res.

4.2.2 Конструктор(ы)

4.2.2.1 calc()

```
calc::calc (
            std::vector< uint32_t > chisla )
```

Конструктор для вычисления суммы элементов вектора

Аргументы

in	chisla	Вектор данных. Не должен быть пустым.

Исключения

no_crit_err,если	вектор пуст.
boost::numeric::positive_overflow,если	происходит переполнение.

4.2.3 Методы

4.2.3.1 send_res()

uint32_t calc::send_res ( )

Метод для отправки результата

Возвращает

Результат вычислений.

Объявления и описания членов классов находятся в файлах:

- calc.h
- calc.cpp

4.3 Класс communicator

Класс коммуникатора

#include <communicator.h>

Открытые члены

- int connection (int port, std::map< std::string, std::string > base, logger *l)
  Соединение с сервером
- std::string sha224 (std::string input_str)
  Формирование хэша методом SHA224.

4.3.1 Подробное описание

Класс коммуникатора

Устанавливает соединение с сервером, производит авторизацию клиента. В качестве метода хэширования выбран SHA224.

4.3.2 Методы

4.3.2.1 connection()

```
int communicator::connection (
                int port,
                std::map< std::string, std::string > base,
                logger * l )
```

Соединение с сервером

Производит соединение с сервером, авторизует пользователя. Передает вектор с данными для вычисления в класс calc.

Аргументы

in	port	Номер порта.
in	base	Контейнер с базой данных.
in	l	Указатель на объект logger для записи всех событий в журнал.

Исключения

crit_err,если	произошел сбой на этапе соединения с сервером, на этапе авторизации или отправки данных.

4.3.2.2 sha224()

```
std::string communicator::sha224 (
                std::string input_str )
```

Формирование хэша методом SHA224.

Производит формирование хэша методом SHA224 библиотеки Crypto++. Формирует хэш соли и пароля.

Аргументы

in	input_str	Входная строка для хэширования.

Возвращает

Результат хэширования.

Объявления и описания членов классов находятся в файлах:

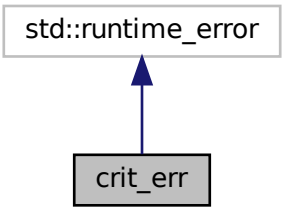
- communicator.h
- communicator.cpp

4.4 Класс crit_err

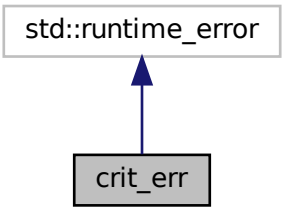
Класс для возбуждения критических ошибок Возбуждает критические ошибки.

#include <error.h>

Граф наследования:crit_err:



Граф связей класса crit_err:



Открытые члены

- crit_err (const std::string &s)

4.4.1 Подробное описание

Класс для возбуждения критических ошибок Возбуждает критические ошибки.

Объявления и описания членов класса находятся в файле:

- error.h

4.5 Класс interface

Класс интерфейса

#include <interface.h>

Открытые члены

- interface ()
  Конструктор по умолчанию
- bool parser (int argc, const char **argv)
  Парсер
- void setup_connection (const std::string &basefile, const std::string &logfile)
  Установка соединения с базой данных и журналом лога
- void spravka (const boost::program_options::options_description &opts)
  Справка
- int get_port () const
  Получение порта
- std::string get_base ()
  Получение пути к базе данных
- std::string get_log ()
  Получение пути к журналу лога

Закрытые данные

- int port
  Переменная с номером порта
- string basefile
  Путь к файлу базы данных
- string logfile
  Путь к файлу журнала

4.5.1 Подробное описание

Класс интерфейса

Получает порт, по умолчанию 33333. Парсер выполняет чтение операндов командной строки. Устанавливается соединение с базой данных и журналом лога. Выполняется вызов справки.

4.5.2 Методы

4.5.2.1 get_base()

std::string interface::get_base ( ) [inline]

Получение пути к базе данных

Возвращает

Путь к базе данных.

4.5.2.2 get_log()

std::string interface::get_log ( ) [inline]

Получение пути к журналу лога

Возвращает

Путь к журналу лога.

4.5.2.3 get_port()

int interface::get_port ( ) const [inline]

Получение порта

Этот метод передает значение порта в коммуникатор.

Возвращает

Номер порта.

4.5.2.4 parser()

```
bool interface::parser (
                int argc,
                const char ** argv )
```

Парсер

Читает операнды командной строки. В случае передачи операнда -h производится вызов справки.

Аргументы

in	argc	Количество операндов.
in	argv	Значение операндов.

Исключения

crit_err	в случае передачи некорректного значения порта.

Возвращает

true или false в случае корректной или некорректной передачи аргументов.

4.5.2.5 setup_connection()

```
void interface::setup_connection (
                const std::string & basefile,
                const std::string & logfile )
```

Установка соединения с базой данных и журналом лога

Устанавливает соединение с базой данных и журналом лога.

Аргументы

in	basefile	Путь к файлу базы данных.
in	logfile	Путь к файлу журнала лога.

4.5.2.6 spravka()

```
void interface::spravka (
                const boost::program_options::options_description & opts )
```

Справка

Вызов справки.

Аргументы

in	opts	Описание опций для справки.

Объявления и описания членов классов находятся в файлах:

- interface.h

- interface.cpp

4.6 Класс logger

Класс для журнала лога

#include <log.h>

Открытые члены

- logger ()
  Конструктор по умолчанию
- logger (const std::string &path, bool log_to_console=false)
  Конструктор с параметром
- int set_path (const std::string &path_file, bool log_to_console=false)
  Установка пути к файлу лога
- int writelog (const std::string &message)
  Запись события в журнал
- std::string gettime ()
  Получение текущего времени
- std::string get_path () const
  Получение пути к файлу лога (для модульного тестирования)

Закрытые данные

- std::string path_to_logfile
  Путь к файлу лога
- bool log_to_console
  Флаг для вывода в консоль

4.6.1 Подробное описание

Класс для журнала лога

4.6.2 Методы

4.6.2.1 get_path()

std::string logger::get_path ( ) const

Получение пути к файлу лога (для модульного тестирования)

Возвращает

Путь к файлу лога.

4.6.2.2 gettime()

```
std::string logger::gettime ( )
```

Получение текущего времени

Позволяет получить время вместе с датой.

Возвращает

Строка с текущим временем.

4.6.2.3 set_path()

```
int logger::set_path (
        const std::string & path_file,
        bool log_to_console = false )
```

Установка пути к файлу лога

Устанавливает путь к файлу лога.

Аргументы

in	path_file	Путь к файлу лога.
in	log_to_console	Флаг для вывода в консоль.

Исключения

crit_err	Если файл не открывается.

4.6.2.4 writelog()

```
int logger::writelog (
        const std::string & message )
```

Запись события в журнал

Записывает событие в лог.

Аргументы

in	message	Сообщение для записи.

Исключения

crit_err	Если файл не открывается на запись.

Объявления и описания членов классов находятся в файлах:

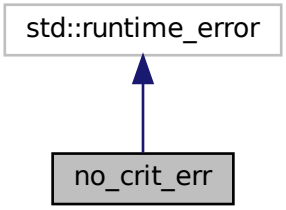
- log.h
- log.cpp

4.7 Класс no_crit_err

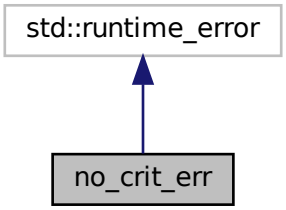
Класс для возбуждения некритических ошибок Возбуждает некритические ошибки.

#include <error.h>

Граф наследования:no_crit_err:



Граф связей класса no_crit_err:



Открытые члены

- no_crit_err (const std::string s)

4.7.1 Подробное описание

Класс для возбуждения некритических ошибок Возбуждает некритические ошибки.

Объявления и описания членов класса находятся в файле:

- error.h

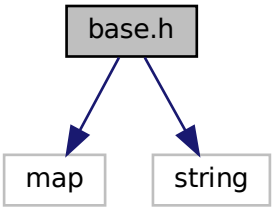
Глава 5

Файлы

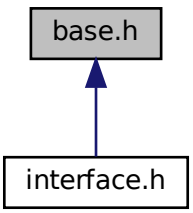
5.1 Файл base.h

Заголовочный файл для модуля базы данных

#include <map>
#include <string>
Граф включаемых заголовочных файлов для base.h:



Граф файлов, в которые включается этот файл:



Классы

- class base

  Класс для чтения базы данных

5.1.1 Подробное описание

Заголовочный файл для модуля базы данных

Автор

Шурманов И.С.

Версия

1.0

Дата

17.12.2024

5.2 base.h

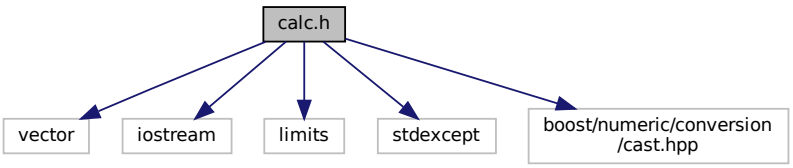
См. документацию.

```
1 #pragma once
2 #include <map>
3 #include <string>
4
16 class base
17 {
18 private:
19     std::map<std::string, std::string> data_base;
20
21 public:
30     void connect(std::string f);
31
36     std::map<std::string, std::string> get_data()
37     {
38         return data_base;
39     }
40
46     bool has_login(const std::string& login)const
47 {
48         return data_base.find(login) != data_base.end();
49     }
50
56     std::string get_password(const std::string& login)const
57 {
58         auto it = data_base.find(login);
59         if (it != data_base.end()) {
60             return it->second;
61         }
62         return "";
63     }
64 };
```

5.3 Файл calc.h

Заголовочный файл для модуля вычислений

#include <vector>
#include <iostream>
#include <limits>
#include <stdexcept>
#include <boost/numeric/conversion/cast.hpp>
Граф включаемых заголовочных файлов для calc.h:



Классы

- class calc

  Класс для вычисления суммы элементов вектора

5.3.1 Подробное описание

Заголовочный файл для модуля вычислений

Автор

Шурманов И.С.

Версия

1.0

Дата

17.12.2024

5.4 calc.h

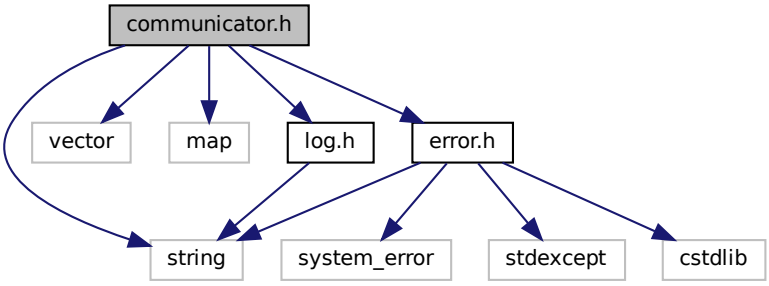
См. документацию.

```
1 #pragma once
2 #include <vector>
3 #include <iostream>
4 #include <limits>
5 #include <stdexcept> // Для исключений
6 #include <boost/numeric/conversion/cast.hpp> // Для проверки на переполнение
7
19 class calc {
20 public:
21     uint32_t res;
22
29     calc(std::vector<uint32_t> chisla);
30
35     uint32_t send_res();
36 };
```

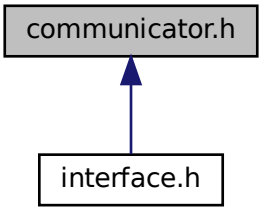
5.5 Файл communicator.h

Заголовочный файл для коммуникатора сервера

#include <string>
#include <vector>
#include <map>
#include "log.h"
#include "error.h"
Граф включаемых заголовочных файлов для communicator.h:



Граф файлов, в которые включается этот файл:



Классы

- class communicator

  Класс коммуникатора

5.5.1 Подробное описание

Заголовочный файл для коммуникатора сервера

Автор

Шурманов И.С.

Версия

1.0

Дата

17.12.2024

5.6 communicator.h

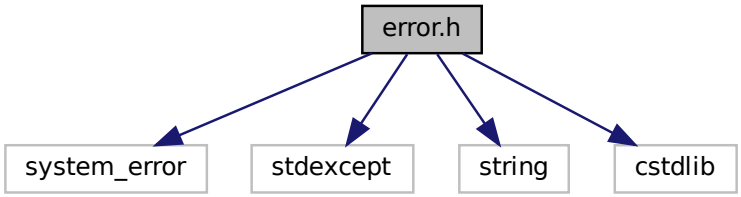
См. документацию.

```
1 #pragma once
2 #include <string>
3 #include <vector>
4 #include <map>
5 #include "log.h"
6 #include "error.h"
7
8 using namespace std;
9
21 class communicator
22 {
23 public:
34     int connection(int port, std::map<std::string, std::string> base, logger* l);
35
44     std::string sha224(std::string input_str);
45 };
```

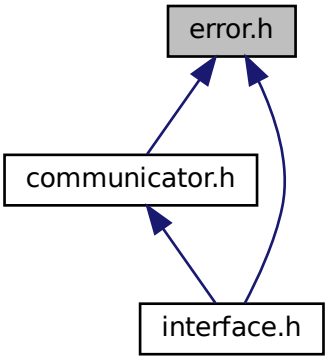
5.7 Файл error.h

Заголовочный файл модуля возбуждения ошибок

#include <system_error>
#include <stdexcept>
#include <string>
#include <cstdlib>
Граф включаемых заголовочных файлов для error.h:



Граф файлов, в которые включается этот файл:



Классы

- class crit_err

  Класс для возбуждения критических ошибок Возбуждает критические ошибки.
- class no_crit_err

  Класс для возбуждения некритических ошибок Возбуждает некритические ошибки.

5.7.1 Подробное описание

Заголовочный файл модуля возбуждения ошибок

Автор

Шурманов И.С.

Версия

1.0

Дата

17.12.2024

5.8 error.h

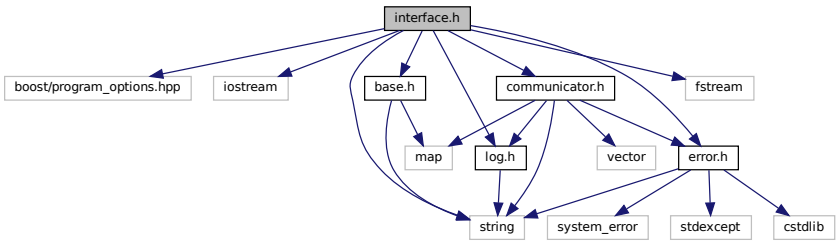
См. документацию.

```
1 #pragma once
2 #include <system_error>
3 #include <stdexcept>
4 #include <string>
5 #include <cstdlib>
6
17 class crit_err : public std::runtime_error
18 {
19 public:
20     crit_err(const std::string& s) : std::runtime_error(s) {}
21 };
22
26 class no_crit_err : public std::runtime_error
27 {
28 public:
29     no_crit_err(const std::string s) : std::runtime_error(s) {}
30 };
```

5.9 Файл interface.h

Заголовочный файл для интерфейса

#include <boost/program_options.hpp>
#include <iostream>
#include <string>
#include <fstream>
#include "log.h"
#include "base.h"
#include "communicator.h"
#include "error.h"
Граф включаемых заголовочных файлов для interface.h:



Классы

- class interface

  Класс интерфейса

5.9.1 Подробное описание

Заголовочный файл для интерфейса

Автор

Шурманов И.С.

Версия

1.0

Дата

17.12.2024

5.10 interface.h

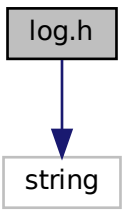
См. документацию.

```
1 #pragma once
2 #include <boost/program_options.hpp>
3 #include <iostream>
4 #include <string>
5 #include <fstream>
6 #include "log.h"
7 #include "base.h"
8 #include "communicator.h"
9 #include "error.h"
10
24 class interface {
25     int port;
26     string basefile;
27     string logfile;
28
29 public:
30     interface() : port(33333), basefile("base.txt"), logfile("log.txt") {}
31
42     bool parser(int argc, const char** argv);
43
51     void setup_connection(const std::string& basefile, const std::string& logfile);
52
59     void spravka(const boost::program_options::options_description& opts);
60
67     int get_port()const { return port; }
68
73     std::string get_base() { return basefile; }
74
79     std::string get_log() { return logfile; }
80 };
```

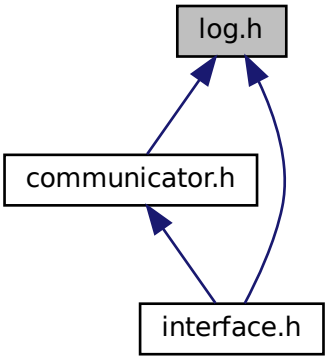
5.11 Файл log.h

Заголовочный файл для установки журнала лога

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



Граф файлов, в которые включается этот файл:



Классы

- class logger

  Класс для журнала лога

5.11.1 Подробное описание

Заголовочный файл для установки журнала лога

Автор

Шурманов И.С.

Версия

1.0

Дата

17.12.2024

5.12 log.h

См. документацию.

```
1 #pragma once
2 #include <string>
3
13 class logger {
14 private:
15     std::string path_to_logfile;
16     bool log_to_console;
17
18 public:
19     logger();
20
21     logger(const std::string& path, bool log_to_console = false);
22
31     int set_path(const std::string& path_file, bool log_to_console = false);
32
40     int writelog(const std::string& message);
41
48     std::string gettime();
49
55     std::string get_path() const;
56 };
```

Предметный указатель