

uint64_t vectors processing server
1.0

Создано системой Doxygen 1.9.1

1 Иерархический список классов	1
1.1 Иерархия классов	1
2 Алфавитный указатель классов	3
2.1 Классы	3
3 Список файлов	5
3.1 Файлы	5
4 Классы	7
4.1 Класс <code>auth_err</code>	7
4.1.1 Подробное описание	8
4.1.2 Конструктор(ы)	8
4.1.2.1 <code>auth_err()</code> [1/2]	8
4.1.2.2 <code>auth_err()</code> [2/2]	8
4.2 Класс <code>getdata</code>	8
4.2.1 Подробное описание	9
4.2.2 Конструктор(ы)	9
4.2.2.1 <code>getdata()</code>	9
4.2.3 Методы	9
4.2.3.1 <code>get()</code>	9
4.2.4 Данные класса	9
4.2.4.1 <code>lp</code>	10
4.3 Класс <code>log_err</code>	10
4.3.1 Подробное описание	11
4.3.2 Конструктор(ы)	11
4.3.2.1 <code>log_err()</code> [1/2]	11
4.3.2.2 <code>log_err()</code> [2/2]	11
4.4 Класс <code>md5h</code>	11
4.4.1 Подробное описание	12
4.4.2 Конструктор(ы)	12
4.4.2.1 <code>md5h()</code>	12
4.4.3 Методы	12
4.4.3.1 <code>hash()</code>	12
4.4.3.2 <code>salt_generator()</code>	12
4.4.4 Данные класса	13
4.4.4.1 <code>salt</code>	13
4.5 Класс <code>read_err</code>	13
4.5.1 Подробное описание	14
4.5.2 Конструктор(ы)	14
4.5.2.1 <code>read_err()</code> [1/2]	14
4.5.2.2 <code>read_err()</code> [2/2]	14
4.6 Класс <code>server</code>	14
4.6.1 Подробное описание	15

4.6.2 Конструктор(ы)	15
4.6.2.1 server()	15
4.6.3 Методы	15
4.6.3.1 accepting_connection()	15
4.6.3.2 authentication()	16
4.6.3.3 handling()	16
4.6.3.4 start()	16
4.6.4 Данные класса	17
4.6.4.1 buf	17
4.6.4.2 lp	17
4.6.4.3 rc	17
4.6.4.4 sckt	17
4.6.4.5 userdata	17
4.6.4.6 wrkr	17
5 Файлы	19
5.1 Файл errorhandler.hpp	19
5.1.1 Подробное описание	20
5.1.2 Функции	20
5.1.2.1 logger()	20
5.2 Файл getdata.cpp	21
5.3 Файл getdata.hpp	21
5.3.1 Подробное описание	22
5.4 Файл main.cpp	22
5.4.1 Подробное описание	23
5.4.2 Макросы	23
5.4.2.1 CRYPTOPP_ENABLE_NAMESPACE_WEAK	23
5.4.3 Функции	23
5.4.3.1 help()	24
5.4.3.2 main()	24
5.4.4 Переменные	24
5.4.4.1 f	24
5.4.4.2 l	24
5.4.4.3 p	24
5.4.4.4	24
5.5 Файл md5hash.cpp	25
5.5.1 Макросы	25
5.5.1.1 CRYPTOPP_ENABLE_NAMESPACE_WEAK	25
5.6 Файл md5hash.hpp	25
5.6.1 Макросы	26
5.6.1.1 CRYPTOPP_ENABLE_NAMESPACE_WEAK	26
5.7 Файл server.cpp	26
5.8 Файл server.hpp	26

5.8.1 Подробное описание	27
5.9 Файл unittest.cpp	27
5.9.1 Функции	28
5.9.1.1 main()	28
5.9.1.2 SUITE() [1/4]	28
5.9.1.3 SUITE() [2/4]	28
5.9.1.4 SUITE() [3/4]	28
5.9.1.5 SUITE() [4/4]	28
Предметный указатель	29

Глава 1

Иерархический список классов

1.1 Иерархия классов

Иерархия классов.

getdata	8
std::invalid_argument	
auth_err	7
read_err	13
md5h	11
std::runtime_error	
log_err	10
server	14

Глава 2

Алфавитный указатель классов

2.1 Классы

Классы с их кратким описанием.

auth_err	Класс обработки ошибок аутентификации	7
getdata	Класс, обрабатывающий файл с ID:PASS клиентов	8
log_err	Класс обработки ошибок записи в журнал	10
md5h	Класс, обрабатывающий HASH MD5	11
read_err	Класс обработки ошибок чтения	13
server	Класс управления работой сервера	14

Глава 3

Список файлов

3.1 Файлы

Полный список файлов.

errorhandler.hpp	
Класс обработки ошибок с записью в журнал	19
getdata.cpp	21
getdata.hpp	
Класс, обрабатывающий файл с ID:PASS клиентов	21
main.cpp	
Функция вывода справки и точка входа в программу	22
md5hash.cpp	25
md5hash.hpp	25
server.cpp	26
server.hpp	
Класс, управляющий работой сервера	26
unittest.cpp	27

Глава 4

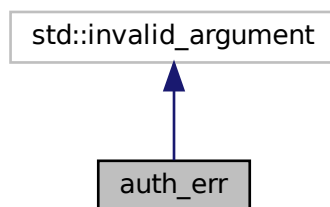
Классы

4.1 Класс `auth_err`

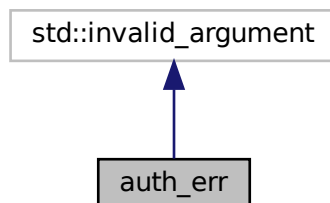
Класс обработки ошибок аутентификации

```
#include <errorhandler.hpp>
```

Граф наследования: `auth_err`:



Граф связей класса `auth_err`:



Открытые члены

- [auth_err](#) (const std::string &s)
- [auth_err](#) (const char *s)

4.1.1 Подробное описание

Класс обработки ошибок аутентификации

4.1.2 Конструктор(ы)

4.1.2.1 [auth_err\(\)](#) [1/2]

```
auth_err::auth_err (  
    const std::string & s ) [inline]
```

4.1.2.2 [auth_err\(\)](#) [2/2]

```
auth_err::auth_err (  
    const char * s ) [inline]
```

Объявления и описания членов класса находятся в файле:

- [errorhandler.hpp](#)

4.2 Класс [getdata](#)

Класс, обрабатывающий файл с ID:PASS клиентов

```
#include <getdata.hpp>
```

Открытые члены

- [getdata](#) (const string lpath)
Конструктор класса
- `map< string, string >` [get](#) (string fpath)
Запись ID:PASS в словарь

Закрытые данные

- string [lp](#)
Путь к файлу с журналом

4.2.1 Подробное описание

Класс, обрабатывающий файл с ID:PASS клиентов

4.2.2 Конструктор(ы)

4.2.2.1 getdata()

```
getdata::getdata (
    const string lpath )
```

Конструктор класса

Аргументы

lpath	Путь к файлу с журналом
-------	-------------------------

4.2.3 Методы

4.2.3.1 get()

```
map< string, string > getdata::get (
    string fpath )
```

Запись ID:PASS в словарь

Аргументы

fpath	Путь к файлу с ID:PASS клиентов
-------	---------------------------------

Возвращает

userdata ID:PASS

4.2.4 Данные класса

4.2.4.1 lp

```
string getdata::lp [private]
```

Путь к файлу с журналом

Объявления и описания членов классов находятся в файлах:

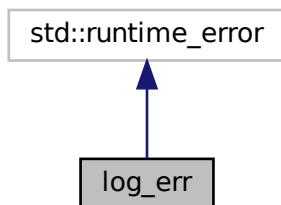
- [getdata.hpp](#)
- [getdata.cpp](#)

4.3 Класс log_err

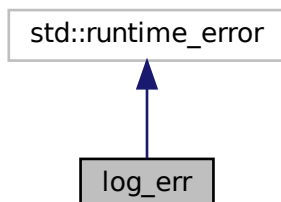
Класс обработки ошибок записи в журнал

```
#include <errorhandler.hpp>
```

Граф наследования:log_err:



Граф связей класса log_err:



Открытые члены

- [log_err](#) (const std::string &s)
- [log_err](#) (const char *s)

4.3.1 Подробное описание

Класс обработки ошибок записи в журнал

4.3.2 Конструктор(ы)

4.3.2.1 log_err() [1/2]

```
log_err::log_err (  
    const std::string & s ) [inline]
```

4.3.2.2 log_err() [2/2]

```
log_err::log_err (  
    const char * s ) [inline]
```

Объявления и описания членов класса находятся в файле:

- [errorhandler.hpp](#)

4.4 Класс md5h

Класс, обрабатывающий HASH MD5.

```
#include <md5hash.hpp>
```

Открытые члены

- [md5h](#) ()
Конструктор класса
- string [hash](#) (const string password)
Получение хэша от пароля

Открытые атрибуты

- string [salt](#)
Соль

Закрытые члены

- string `salt_generator` ()
Генератор соли

4.4.1 Подробное описание

Класс, обрабатывающий HASH MD5.

4.4.2 Конструктор(ы)

4.4.2.1 md5h()

`md5h::md5h ()`

Конструктор класса

4.4.3 Методы

4.4.3.1 hash()

`string md5h::hash (`
`const string password)`

Получение хэша от пароля

Аргументы

<code>password</code>	Пароль
-----------------------	--------

Возвращает

digest Хэш от пароля

4.4.3.2 salt_generator()

`string md5h::salt_generator () [private]`

Генератор соли

Возвращает

tmp_s Соль

4.4.4 Данные класса

4.4.4.1 salt

string md5h::salt

Соль

Объявления и описания членов классов находятся в файлах:

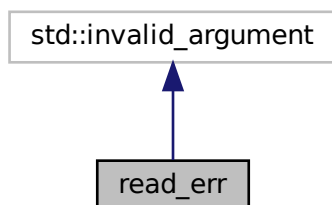
- [md5hash.hpp](#)
- [md5hash.cpp](#)

4.5 Класс read_err

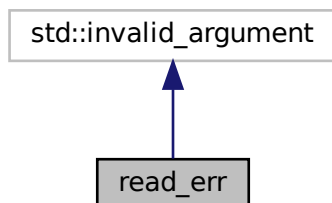
Класс обработки ошибок чтения

```
#include <errorhandler.hpp>
```

Граф наследования:read_err:



Граф связей класса read_err:



Открытые члены

- [read_err](#) (const std::string &s)
- [read_err](#) (const char *s)

4.5.1 Подробное описание

Класс обработки ошибок чтения

4.5.2 Конструктор(ы)

4.5.2.1 read_err() [1/2]

```
read_err::read_err (
    const std::string & s ) [inline]
```

4.5.2.2 read_err() [2/2]

```
read_err::read_err (
    const char * s ) [inline]
```

Объявления и описания членов класса находятся в файле:

- [errorhandler.hpp](#)

4.6 Класс server

Класс управления работой сервера

```
#include <server.hpp>
```

Открытые члены

- [server](#) (std::string fpath, int port, std::string lpath)
Конструктор класса
- int [start](#) (int port, const char *address)
Функция запуска сервера и привязки порта
- int [accepting_connection](#) ()
Функция ожидания и принятия соединения с клиентом
- bool [authentication](#) ()
Функция аутентификации пользователя
- bool [handling](#) ()
Функция обработки векторов

Закрытые данные

- `std::string lp`
Путь к файлу с логом
- `int sckt`
Сокет
- `int wrkr`
Действующее соединение с клиентом
- `int rc`
Ответ от клиента
- `char * buf = new char[4096]`
Буфер
- `std::map< std::string, std::string > userdata`
Данные клиента ID:PASS.

4.6.1 Подробное описание

Класс управления работой сервера

4.6.2 Конструктор(ы)

4.6.2.1 server()

```
server::server (  
    std::string fpath,  
    int port,  
    std::string lpath )
```

Конструктор класса

Аргументы

fpath	Путь к файлу с ID:PASS клиентов
port	Порт для приема соединения
lpath	Путь к файлу журнала

4.6.3 Методы

4.6.3.1 accepting_connection()

```
int server::accepting_connection ( )
```

Функция ожидания и принятия соединения с клиентом

Возвращает

wrkr Действующее соединение с клиентом

4.6.3.2 authentication()

```
bool server::authentication ( )
```

Функция аутентификации пользователя

Возвращает

True, если аутентификация удалась, False - если нет

4.6.3.3 handling()

```
bool server::handling ( )
```

Функция обработки векторов

Возвращает

True, если обработка удалась, False - если нет

4.6.3.4 start()

```
int server::start (
    int port,
    const char * address )
```

Функция запуска сервера и привязки порта

Аргументы

port	Порт для приема соединения
address	Сетевой адрес для приема соединения

Возвращает

sckt Сокет

4.6.4 Данные класса

4.6.4.1 buf

```
char* server::buf = new char[4096] [private]
```

Буфер

4.6.4.2 lp

```
std::string server::lp [private]
```

Путь к файлу с логом

4.6.4.3 rc

```
int server::rc [private]
```

Ответ от клиента

4.6.4.4 sckt

```
int server::sckt [private]
```

Сокет

4.6.4.5 userdata

```
std::map<std::string, std::string> server::userdata [private]
```

Данные клиента ID:PASS.

4.6.4.6 wrkr

```
int server::wrkr [private]
```

Действующее соединение с клиентом

Объявления и описания членов классов находятся в файлах:

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

Глава 5

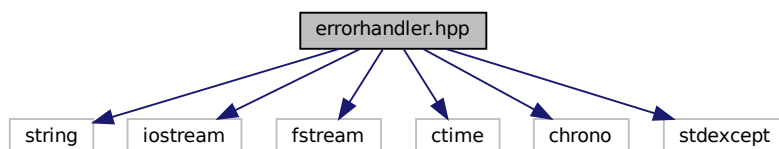
Файлы

5.1 Файл errorhandler.hpp

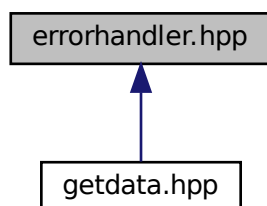
Класс обработки ошибок с записью в журнал

```
#include <string>
#include <iostream>
#include <fstream>
#include <ctime>
#include <chrono>
#include <stdexcept>
```

Граф включаемых заголовочных файлов для errorhandler.hpp:



Граф файлов, в которые включается этот файл:



Классы

- class `log_err`
Класс обработки ошибок записи в журнал
- class `auth_err`
Класс обработки ошибок аутентификации
- class `read_err`
Класс обработки ошибок чтения

Функции

- `std::string logger` (`std::string lp`, `std::string what`)
Функция записи ошибок в журнал

5.1.1 Подробное описание

Класс обработки ошибок с записью в журнал

Автор

Малицкий М.А.

Версия

1.0

Дата

21.12.2023

Авторство

ИБСТ ПГУ

5.1.2 Функции

5.1.2.1 `logger()`

```
std::string logger (  
    std::string lp,  
    std::string what ) [inline]
```

Функция записи ошибок в журнал

Аргументы

lp	Путь к файлу журнала
what	Содержание ошибки

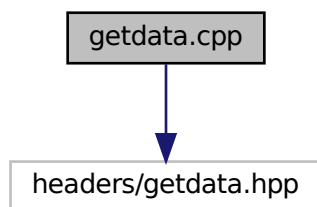
Возвращает

log_s Строка с логом ошибки

5.2 Файл getdata.cpp

```
#include "headers/getdata.hpp"
```

Граф включаемых заголовочных файлов для getdata.cpp:

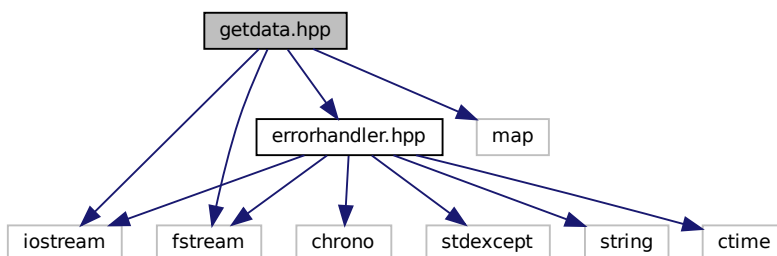


5.3 Файл getdata.hpp

Класс, обрабатывающий файл с ID:PASS клиентов

```
#include <iostream>
#include <fstream>
#include <map>
#include "errorhandler.hpp"
```

Граф включаемых заголовочных файлов для getdata.hpp:



Классы

- class [getdata](#)

Класс, обрабатывающий файл с ID:PASS клиентов

5.3.1 Подробное описание

Класс, обрабатывающий файл с ID:PASS клиентов

Автор

Малицкий М.А.

Версия

1.0

Дата

21.12.2023

Авторство

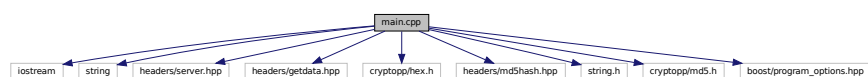
ИБСТ ПГУ

5.4 Файл main.cpp

Функция вывода справки и точка входа в программу

```
#include <iostream>
#include <string>
#include "headers/server.hpp"
#include "headers/getdata.hpp"
#include <cryptopp/hex.h>
#include "headers/md5hash.hpp"
#include <string.h>
#include <cryptopp/md5.h>
#include <boost/program_options.hpp>
```

Граф включаемых заголовочных файлов для main.cpp:



Макросы

- #define [CRYPTOPP_ENABLE_NAMESPACE_WEAK](#) 1

Функции

- void `help` ()
- int `main` (int argc, const char *argv[])

Переменные

- struct {
 uint32_t `p` = 33333
 std::string `f` = "database.txt"
 std::string `l` = "log.txt"
} `params`

5.4.1 Подробное описание

Функция вывода справки и точка входа в программу

Автор

Малицкий М.А.

Версия

1.0

Дата

21.12.2023

Авторство

ИБСТ ПГУ

5.4.2 Макросы

5.4.2.1 CRYPTOPP_ENABLE_NAMESPACE_WEAK

```
#define CRYPTOPP_ENABLE_NAMESPACE_WEAK 1
```

5.4.3 Функции

5.4.3.1 help()

```
void help ( )
```

5.4.3.2 main()

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

5.4.4 Переменные

5.4.4.1 f

```
std::string f ="database.txt"
```

5.4.4.2 l

```
std::string l ="log.txt"
```

5.4.4.3 p

```
uint32_t p =33333
```

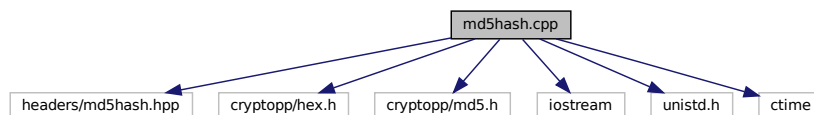
5.4.4.4

```
struct { ... } params
```

5.5 Файл md5hash.cpp

```
#include "headers/md5hash.hpp"
#include <cryptopp/hex.h>
#include <cryptopp/md5.h>
#include <iostream>
#include <unistd.h>
#include <ctime>
```

Граф включаемых заголовочных файлов для md5hash.cpp:



Макросы

- `#define CRYPTOPP_ENABLE_NAMESPACE_WEAK 1`

5.5.1 Макросы

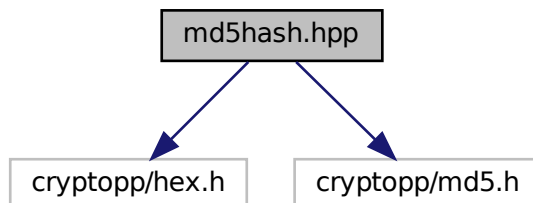
5.5.1.1 CRYPTOPP_ENABLE_NAMESPACE_WEAK

```
#define CRYPTOPP_ENABLE_NAMESPACE_WEAK 1
```

5.6 Файл md5hash.hpp

```
#include <cryptopp/hex.h>
#include <cryptopp/md5.h>
```

Граф включаемых заголовочных файлов для md5hash.hpp:



Классы

- class `md5h`
Класс, обрабатывающий HASH MD5.

Макросы

- `#define CRYPTOPP_ENABLE_NAMESPACE_WEAK 1`

5.6.1 Макросы

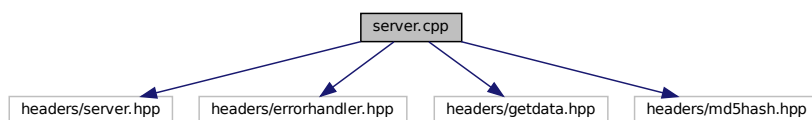
5.6.1.1 CRYPTOPP_ENABLE_NAMESPACE_WEAK

```
#define CRYPTOPP_ENABLE_NAMESPACE_WEAK 1
```

5.7 Файл server.cpp

```
#include "headers/server.hpp"
#include "headers/errorhandler.hpp"
#include "headers/getdata.hpp"
#include "headers/md5hash.hpp"
```

Граф включаемых заголовочных файлов для server.cpp:



5.8 Файл server.hpp

Класс, управляющий работой сервера

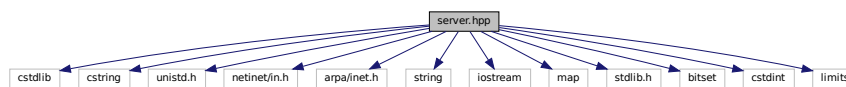
```
#include <cstdlib>
#include <cstring>
#include <unistd.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <string>
#include <iostream>
#include <map>
#include <stdlib.h>
#include <bitset>
```



```
#include <cstdint>
```

```
#include <limits>
```

Граф включаемых заголовочных файлов для server.hpp:



Классы

- class [server](#)

Класс управления работой сервера

5.8.1 Подробное описание

Класс, управляющий работой сервера

Автор

Малицкий М.А.

Версия

1.0

Дата

21.12.2023

Авторство

ИБСТ ПГУ

5.9 Файл unittest.cpp

```
#include <UnitTest++/UnitTest++.h>
```

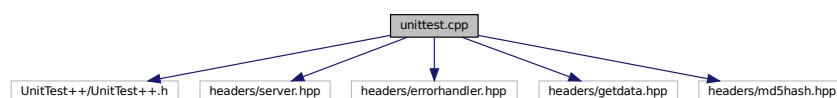
```
#include "headers/server.hpp"
```

```
#include "headers/errorhandler.hpp"
```

```
#include "headers/getdata.hpp"
```

```
#include "headers/md5hash.hpp"
```

Граф включаемых заголовочных файлов для unittest.cpp:



Функции

- [SUITE](#) (ServerTests)
- [SUITE](#) (GetDataTests)
- [SUITE](#) (HashTest)
- [SUITE](#) (ErrorLoggingTest)
- `int main` (int argc, char **argv)

5.9.1 Функции

5.9.1.1 main()

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

5.9.1.2 SUITE() [1/4]

```
SUITE (  
    ErrorLoggingTest )
```

5.9.1.3 SUITE() [2/4]

```
SUITE (  
    GetDataTests )
```

5.9.1.4 SUITE() [3/4]

```
SUITE (  
    HashTest )
```

5.9.1.5 SUITE() [4/4]

```
SUITE (  
    ServerTests )
```

Предметный указатель

- accepting_connection
 - server, [15](#)
- auth_err, [7](#)
 - auth_err, [8](#)
- authentication
 - server, [16](#)
- buf
 - server, [17](#)
- CRYPTOPP_ENABLE_NAMESPACE_WEAK
 - main.cpp, [23](#)
 - md5hash.cpp, [25](#)
 - md5hash.hpp, [26](#)
- errorhandler.hpp, [19](#)
 - logger, [20](#)
- f
 - main.cpp, [24](#)
- get
 - getdata, [9](#)
- getdata, [8](#)
 - get, [9](#)
 - getdata, [9](#)
 - lp, [9](#)
- getdata.cpp, [21](#)
- getdata.hpp, [21](#)
- handling
 - server, [16](#)
- hash
 - md5h, [12](#)
- help
 - main.cpp, [23](#)
- l
 - main.cpp, [24](#)
- log_err, [10](#)
 - log_err, [11](#)
- logger
 - errorhandler.hpp, [20](#)
- lp
 - getdata, [9](#)
 - server, [17](#)
- main
 - main.cpp, [24](#)
 - unittest.cpp, [28](#)
- main.cpp, [22](#)
- CRYPTOPP_ENABLE_NAMESPACE_WEAK,
 - [23](#)
- f, [24](#)
- help, [23](#)
- l, [24](#)
- main, [24](#)
- p, [24](#)
- params, [24](#)
- md5h, [11](#)
 - hash, [12](#)
 - md5h, [12](#)
 - salt, [13](#)
 - salt_generator, [12](#)
- md5hash.cpp, [25](#)
 - CRYPTOPP_ENABLE_NAMESPACE_WEAK,
 - [25](#)
- md5hash.hpp, [25](#)
 - CRYPTOPP_ENABLE_NAMESPACE_WEAK,
 - [26](#)
- p
 - main.cpp, [24](#)
- params
 - main.cpp, [24](#)
- rc
 - server, [17](#)
- read_err, [13](#)
 - read_err, [14](#)
- salt
 - md5h, [13](#)
- salt_generator
 - md5h, [12](#)
- sckt
 - server, [17](#)
- server, [14](#)
 - accepting_connection, [15](#)
 - authentication, [16](#)
 - buf, [17](#)
 - handling, [16](#)
 - lp, [17](#)
 - rc, [17](#)
 - sckt, [17](#)
 - server, [15](#)
 - start, [16](#)
 - userdata, [17](#)
 - wrkr, [17](#)
- server.cpp, [26](#)
- server.hpp, [26](#)

start
 server, [16](#)
SUITE
 unittest.cpp, [28](#)

unittest.cpp, [27](#)
 main, [28](#)
 SUITE, [28](#)
userdata
 server, [17](#)

wrkr
 server, [17](#)