

## My Project

Generated by Doxygen 1.9.4



---

<b>1 Hierarchical Index</b>	<b>1</b>
1.1 Class Hierarchy . . . . .	1
<b>2 Class Index</b>	<b>3</b>
2.1 Class List . . . . .	3
<b>3 File Index</b>	<b>5</b>
3.1 File List . . . . .	5
<b>4 Class Documentation</b>	<b>7</b>
4.1 error_server Class Reference . . . . .	7
4.2 Server Class Reference . . . . .	7
<b>5 File Documentation</b>	<b>9</b>
5.1 mdfile.cpp File Reference . . . . .	9
5.1.1 Detailed Description . . . . .	9
5.2 mdfile.h File Reference . . . . .	9
5.2.1 Detailed Description . . . . .	10
5.3 mdfile.h . . . . .	10
<b>Index</b>	<b>13</b>



# Chapter 1

## Hierarchical Index

### 1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

invalid_argument	
error_server . . . . .	<a href="#">7</a>
Server . . . . .	<a href="#">7</a>



## Chapter 2

# Class Index

### 2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">error_server</a> . . . . .	<a href="#">7</a>
<a href="#">Server</a> . . . . .	<a href="#">7</a>





## Chapter 3

# File Index

### 3.1 File List

Here is a list of all documented files with brief descriptions:

<a href="#">mdfile.cpp</a>	File vzaimodeystviya . . . . .	9
<a href="#">mdfile.h</a>	Opisanie Classa <a href="#">Server</a> . . . . .	9



## Chapter 4

# Class Documentation

### 4.1 error\_server Class Reference

Inheritance diagram for error\_server:

### 4.2 Server Class Reference

#### Public Member Functions

- string \* **comstr** (int argc, char \*argv[])
- int **self\_addr** (string error, string file\_error, int port)
- int **client\_addr** (int s, string error, string file\_error)

#### Public Attributes

- int **port**
- string **file\_name**
- string **file\_error**
- string **error**
- string **error\_name**

The documentation for this class was generated from the following file:

- [mdfile.h](#)



## Chapter 5

# File Documentation

### 5.1 mdfile.cpp File Reference

file vzaimodeystviya

#### 5.1.1 Detailed Description

file vzaimodeystviya

### 5.2 mdfile.h File Reference

Opisanie Classa [Server](#).

```
#include <netinet/in.h>
#include <iostream>
#include <cassert>
#include <arpa/inet.h>
#include <cstdlib>
#include <unistd.h>
#include <ctime>
#include <fstream>
#include <sstream>
#include <string>
#include <cryptopp/cryptlib.h>
#include <vector>
#include <tuple>
#include <getopt.h>
#include <cryptopp/hex.h>
#include <cryptopp/md5.h>
Include dependency graph for mdfile.h:
```

#### Classes

- class [error\\_server](#)
- class [Server](#)

## Macros

- `#define CRYPTOPP_ENABLE_NAMESPACE_WEAK 1`

## Functions

- `std::string MD` (`std::string sah`)
- `void errors` (`string error`, `string file_name`, `string file_error`)
- `void msgsend` (`int work_sock`, `string mess`)
- `int authorized` (`int work_sock`, `string file_name`, `string file_error`)
- `int math` (`int work_sock`)

### 5.2.1 Detailed Description

Opisanie Classa [Server](#).

Opisanie Classov u methods.

#### Author

Uchastkin I.A.

#### Version

1.0

#### Date

15.11.20203

#### Copyright

IBST PSU

## 5.3 mdfile.h

[Go to the documentation of this file.](#)

```
1
10 #include <netinet/in.h>
11 #include <iostream>
12 #include <cassert>
13 #include <arpa/inet.h>
14 #include <cstdlib>
15 #include <unistd.h>
16 #include <ctime>
17 #include <fstream>
18 #include <sstream>
19 #include <string>
20 #include <cryptopp/cryptlib.h>
21 #include <iostream>
22 #include <vector>
23 #include <tuple>
24 #include <getopt.h>
25 #include <cryptopp/hex.h> // HexEncoder
26 #define CRYPTOPP_ENABLE_NAMESPACE_WEAK 1
27 using namespace CryptoPP;
28 using namespace std;
```

```
29 #include <cryptopp/md5.h>
30
31 std::string MD(std::string sah);
32
33 void errors(string error, string file_name, string file_error);
34
35 void msgsend(int work_sock, string mess);
36
37 int authorized(int work_sock, string file_name, string file_error);
38
39 int math(int work_sock);
40
41 class error_server:
42     public invalid_argument
43 {
44 public:
45     explicit error_server (const string& what_arg):
46         std::invalid_argument(what_arg){}
47     explicit error_server (const char* what_arg):
48         std::invalid_argument(what_arg){}
49 };
50
51 class Server{
52 private:
53
54 public:
55     int port;
56     string file_name;
57     string file_error;
58     string error;
59     string error_name;
60     string* comstr(int argc, char *argv[]);
61     int self_addr(string error, string file_error, int port);
62     int client_addr(int s, string error, string file_error);
63 };
64
65
```





# Index

`error_server`, [7](#)

`mdfile.cpp`, [9](#)

`mdfile.h`, [9](#)

`Server`, [7](#)