

# Документация на использование сервера

## 1.0

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 <code>DataReader</code>	5
3.1.1	Подробное описание	5
3.1.2	Методы	5
3.1.2.1	<code>getClient()</code>	5
3.2	Структура <code>DataReader_fix</code>	6
3.3	Класс <code>Errors</code>	6
3.3.1	Подробное описание	7
3.3.2	Методы	7
3.3.2.1	<code>error_recording()</code>	7
3.4	Структура <code>Errors_fix</code>	8
3.5	Структура <code>Server_fix</code>	8
3.6	Класс <code>ServerClientInterface</code>	9
3.6.1	Подробное описание	9
3.6.2	Методы	10
3.6.2.1	<code>interaction()</code>	10
3.6.2.2	<code>sum_vec()</code>	10
3.7	Класс <code>User</code>	10
3.7.1	Подробное описание	11
3.7.2	Методы	11
3.7.2.1	<code>CheckPassword()</code>	11
3.7.2.2	<code>CheckLogin()</code>	11
3.8	Структура <code>User_fix</code>	12
4	Файлы	13
4.1	Файл <code>DataReader.cpp</code>	13
4.2	Файл <code>DataReader.h</code>	13
4.2.1	Подробное описание	14
4.3	Файл <code>Errors.cpp</code>	15
4.4	Файл <code>Errors.h</code>	15
4.4.1	Подробное описание	16
4.5	Файл <code>main.cpp</code>	17
4.5.1	Подробное описание	17
4.6	Файл <code>md5.cpp</code>	17
4.6.1	Функции	18
4.6.1.1	<code>MD5_hash()</code>	18
4.7	Файл <code>ServerClientInterface.cpp</code>	18
4.8	Файл <code>ServerClientInterface.h</code>	18

4.8.1 Подробное описание . . . . .	19
4.9 Файл User.cpp . . . . .	20
4.10 Файл User.h . . . . .	20
4.10.1 Подробное описание . . . . .	21
Предметный указатель . . . . .	23

# Глава 1

## Алфавитный указатель классов

### 1.1 Классы

Классы с их кратким описанием.

<a href="#">DataReader</a>		
	Класс для получения БД клиентов . . . . .	5
<a href="#">DataReader_fix</a>	. . . . .	6
<a href="#">Errors</a>		
	Класс обработки ошибок . . . . .	6
<a href="#">Errors_fix</a>	. . . . .	8
<a href="#">Server_fix</a>	. . . . .	8
<a href="#">ServerClientInterface</a>		
	Класс для взаимодействия сервера с клиентами . . . . .	9
<a href="#">User</a>		
	Класс, представляющий информацию о подключенном пользователе . . . . .	10
<a href="#">User_fix</a>	. . . . .	12



## Глава 2

# Список файлов

### 2.1 Файлы

Полный список документированных файлов.

<a href="#">DataReader.cpp</a>	13
<a href="#">DataReader.h</a>	
Заголовочный файл для модуля <a href="#">DataReader</a>	13
<a href="#">Errors.cpp</a>	15
<a href="#">Errors.h</a>	
Заголовочный файл для модуля <a href="#">Errors</a>	15
<a href="#">main.cpp</a>	
Функция для предоставления пользователю справки по использованию программы	17
<a href="#">md5.cpp</a>	17
<a href="#">md5.h</a>	??
<a href="#">ServerClientInterface.cpp</a>	18
<a href="#">ServerClientInterface.h</a>	
Заголовочный файл для модуля <a href="#">ServerClientInterface</a>	18
<a href="#">User.cpp</a>	20
<a href="#">User.h</a>	
Заголовочный файл для модуля <a href="#">User</a>	20





## Глава 3

# Классы

### 3.1 Класс DataReader

Класс для получения БД клиентов

```
#include <DataReader.h>
```

Открытые члены

- string [get\\_FileReader](#) ()  
Геттер для атрибута FileReader.
- void [set\\_FileReader](#) (string file)  
Сеттер для атрибута FileReader.
- pair< vector< string >, vector< string > > [getClient](#) ()  
Функция для получения БД клиентов

#### 3.1.1 Подробное описание

Класс для получения БД клиентов

Аргументы

FileReader	путь к файлу с БД клиентов
Err	объект класса <a href="#">Errors</a> обработки ошибок

#### 3.1.2 Методы

##### 3.1.2.1 getClient()

```
pair< vector< string >, vector< string > > DataReader::getClient ( )
```

Функция для получения БД клиентов

Возвращает

Возвращает два вектора: логины и пароли клиентов

Исключения

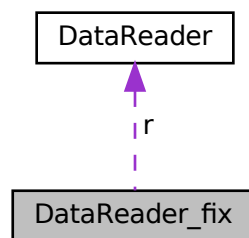
<a href="#">Errors</a>	при возникновении ошибки
------------------------	--------------------------

Объявления и описания членов классов находятся в файлах:

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

## 3.2 Структура DataReader\_fix

Граф связей класса DataReader\_fix:



Открытые атрибуты

- [DataReader](#) \* r

Объявления и описания членов структуры находятся в файле:

- `unit.cpp`

## 3.3 Класс Errors

Класс обработки ошибок

```
#include <Errors.h>
```

## Открытые члены

- `string get_File_Log ()`  
Геттер для атрибута File\_Log.
- `void set_File_Log (string file)`  
Сеттер для атрибута File\_Log.
- `void error_recording (string flag, string info)`  
Функция сохранения ошибок в журнал

### 3.3.1 Подробное описание

#### Класс обработки ошибок

##### Аргументы

File_Log	путь к файлу с журналом ошибок
----------	--------------------------------

### 3.3.2 Методы

#### 3.3.2.1 error\_recording()

```
void Errors::error_recording (  
    string flag,  
    string info )
```

#### Функция сохранения ошибок в журнал

##### Аргументы

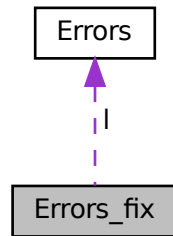
flag	Хранит в себе тип ошибки
info	Хранит в себе подробную информацию об ошибке

Объявления и описания членов классов находятся в файлах:

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

### 3.4 Структура Errors\_fix

Граф связей класса Errors\_fix:



Открытые атрибуты

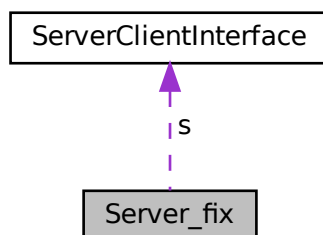
- [Errors](#) \* 1

Объявления и описания членов структуры находятся в файле:

- unit.cpp

### 3.5 Структура Server\_fix

Граф связей класса Server\_fix:



Открытые атрибуты

- [ServerClientInterface](#) \* s

Объявления и описания членов структуры находятся в файле:

- unit.cpp

## 3.6 Класс ServerClientInterface

Класс для взаимодействия сервера с клиентами

```
#include <ServerClientInterface.h>
```

Открытые члены

- int [interaction](#) (string database, string logFile)  
Главная функция для взаимодействия сервера с клиентами
- uint32\_t [sum\\_vec](#) ()  
Функция вычисления суммы векторов
- string [get\\_address](#) ()  
Геттер для атрибута address.
- void [set\\_address](#) (string address1)  
Сеттер для атрибута address.
- int [get\\_port](#) ()  
Геттер для атрибута port.
- void [set\\_port](#) (int port1)  
Сеттер для атрибута port.
- vector< uint32\_t > [get\\_vec](#) ()  
Сеттер для атрибута vec.
- void [set\\_vec](#) (vector< uint32\_t > v)  
Геттер для атрибута vec.
- string [get\\_salt](#) ()  
Геттер для атрибута salt.
- void [set\\_salt](#) ()  
Сеттер для атрибута salt.
- pair< vector< string >, vector< string > > [get\\_DB\\_clients](#) ()  
Геттер для атрибута DB\_clients.
- void [set\\_DB\\_clients](#) (vector< string > login, vector< string > password)  
Сеттер для атрибута DB\_clients.

### 3.6.1 Подробное описание

Класс для взаимодействия сервера с клиентами

Аргументы

address	Адрес сервера
port	Порт сервера
vec	Вектор для вычисления расчетов
DB_clients	БД клиентов
Err	объект класса <a href="#">Errors</a> обработки ошибок

### 3.6.2 Методы

#### 3.6.2.1 interaction()

```
int ServerClientInterface::interaction (
    string database,
    string logFile )
```

Главная функция для взаимодействия сервера с клиентами

Исключения

<a href="#">Errors</a>	при возникновении ошибки
------------------------	--------------------------

#### 3.6.2.2 sum\_vec()

```
uint32_t ServerClientInterface::sum_vec ( )
```

Функция вычисления суммы векторов

Возвращает

Возвращает сумму векторов атрибута vec

Объявления и описания членов классов находятся в файлах:

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

## 3.7 Класс User

Класс, представляющий информацию о подключенном пользователе

```
#include <User.h>
```

Открытые члены

- bool [CheckLogin](#) (vector< string > Db\_ID)  
Функция для проверки ID подключенного клиента
- bool [CheckPassword](#) (vector< string > Db\_hash, vector< string > Db\_ID, string SALT)  
Функция для проверки пароля подключенного клиента
- string [get\\_ID](#) ()  
Геттер для атрибута ID.
- void [set\\_ID](#) (string ID1)  
Сеттер для атрибута ID.
- string [get\\_hash](#) ()  
Геттер для атрибута hash.
- void [set\\_hash](#) (string hash1)  
Сеттер для атрибута hash.

### 3.7.1 Подробное описание

Класс, представляющий информацию о подключенном пользователе

Аргументы

ID	ID подключенного клиента
hash	хэш-код подключенного клиента

### 3.7.2 Методы

#### 3.7.2.1 CheckPassword()

```
bool User::CheckPassword (
    vector< string > Db_password,
    vector< string > Db_ID,
    string SALT )
```

Функция для проверки пароля подключенного клиента

Возвращает

Возвращает true - если пароль подключенного клиента совпадает с паролем из БД, иначе false

#### 3.7.2.2 CheckLogin()

```
bool User::CheckLogin (
    vector< string > Db_ID )
```

Функция для проверки ID подключенного клиента

Возвращает

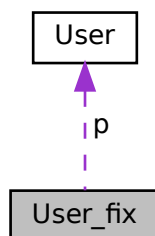
Возвращает true - если в БД есть ID подключенного клиента, иначе false

Объявления и описания членов классов находятся в файлах:

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

### 3.8 Структура User\_fix

Граф связей класса User\_fix:



Открытые атрибуты

- `User * p`

Объявления и описания членов структуры находятся в файле:

- `unit.cpp`



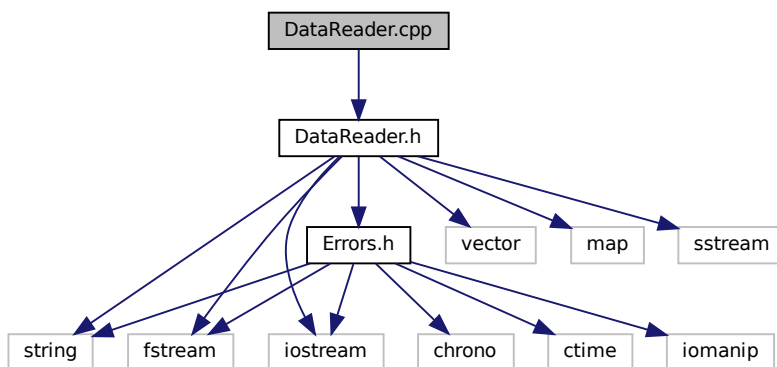
## Глава 4

# Файлы

### 4.1 Файл DataReader.cpp

```
#include "DataReader.h"
```

Граф включаемых заголовочных файлов для DataReader.cpp:



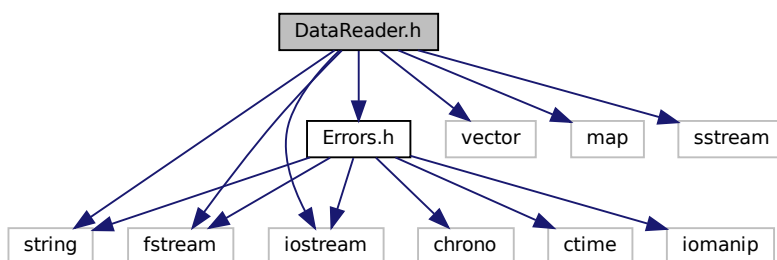
### 4.2 Файл DataReader.h

Заголовочный файл для модуля [DataReader](#).

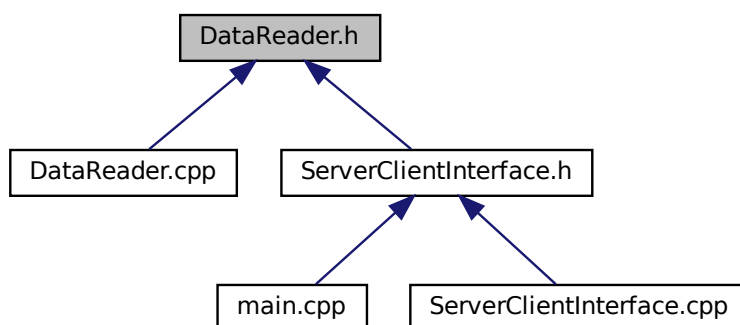
```
#include <string>
#include <vector>
#include <map>
#include <fstream>
#include <sstream>
#include <iostream>
```

```
#include "Errors.h"
```

Граф включаемых заголовочных файлов для `DataReader.h`:



Граф файлов, в которые включается этот файл:



## Классы

- class [DataReader](#)

Класс для получения БД клиентов

### 4.2.1 Подробное описание

Заголовочный файл для модуля [DataReader](#).

Автор

Попов Г.В.

Версия

1.0

Дата

13.12.2024

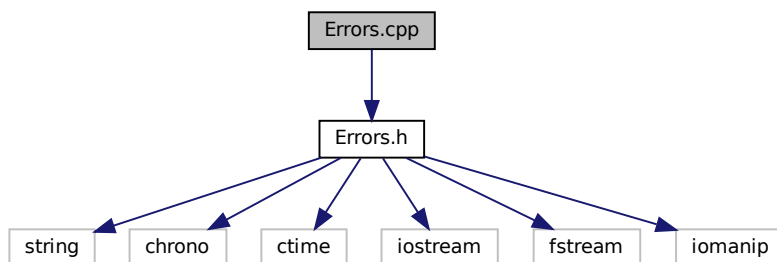
Авторство

ИБСТ ПГУ

## 4.3 Файл Errors.cpp

```
#include "Errors.h"
```

Граф включаемых заголовочных файлов для Errors.cpp:

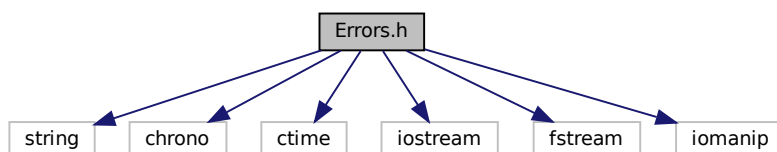


## 4.4 Файл Errors.h

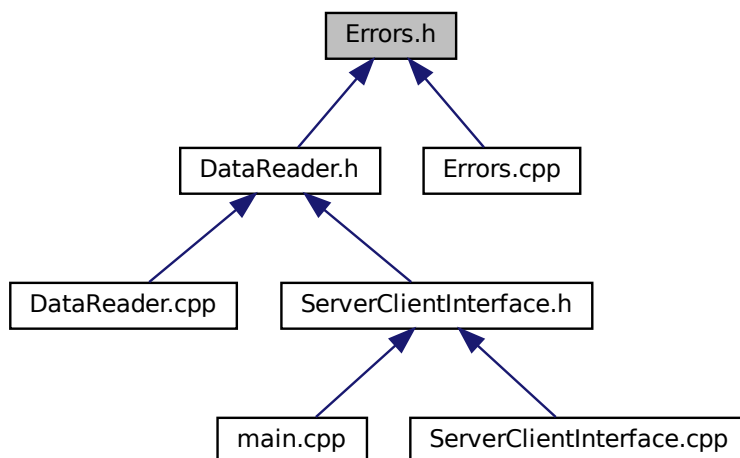
Заголовочный файл для модуля [Errors](#).

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

Граф включаемых заголовочных файлов для Errors.h:



Граф файлов, в которые включается этот файл:



## Классы

- class [Errors](#)  
Класс обработки ошибок

### 4.4.1 Подробное описание

Заголовочный файл для модуля [Errors](#).

Автор

Попов Г.В.

Версия

10.0

Дата

13.12.2023

Авторство

ИБСТ ПГУ

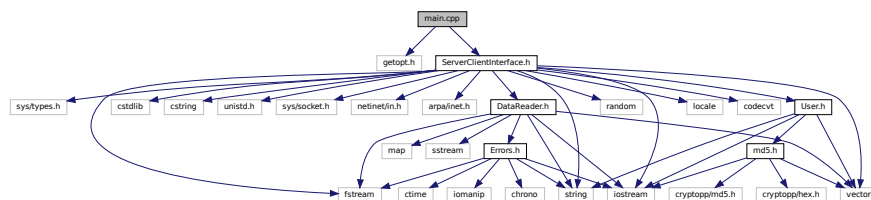
## 4.5 Файл main.cpp

Функция для предоставления пользователю справки по использованию программы

```
#include <getopt.h>
```

```
#include "ServerClientInterface.h"
```

Граф включаемых заголовочных файлов для main.cpp:



### Функции

- void [help](#) ()  
Функция для получения справки по использованию программы
- int [main](#) (int argc, char \*argv[])  
Функция для получения от оператора параметров командной строки

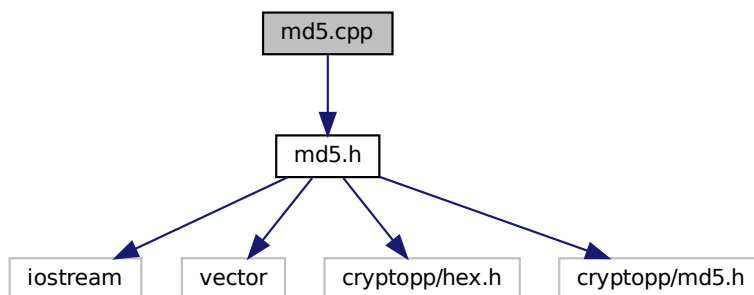
#### 4.5.1 Подробное описание

Функция для предоставления пользователю справки по использованию программы

## 4.6 Файл md5.cpp

```
#include "md5.h"
```

Граф включаемых заголовочных файлов для md5.cpp:



## Функции

- `std::string MD5_hash (std::string msg)`  
Получение хэш-кода по алгоритму md5.

### 4.6.1 Функции

#### 4.6.1.1 MD5\_hash()

```
std::string MD5_hash (
    std::string msg )
```

Получение хэш-кода по алгоритму md5.

Аргументы

msg	сообщения для хэширования
-----	---------------------------

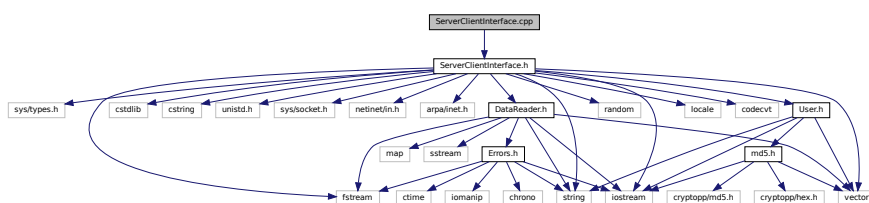
Возвращает

хэш-код

## 4.7 Файл ServerClientInterface.cpp

```
#include "ServerClientInterface.h"
```

Граф включаемых заголовочных файлов для ServerClientInterface.cpp:



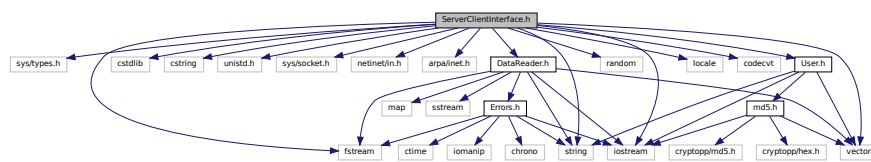
## 4.8 Файл ServerClientInterface.h

Заголовочный файл для модуля [ServerClientInterface](#).

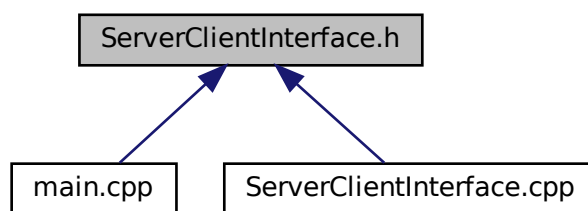
```
#include <sys/types.h>
#include <iostream>
#include <cstdlib>
#include <cstring>
#include <unistd.h>
```

```
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <fstream>
#include <vector>
#include <random>
#include <string>
#include <locale>
#include <codecvt>
#include "DataReader.h"
#include "User.h"
```

Граф включаемых заголовочных файлов для ServerClientInterface.h:



Граф файлов, в которые включается этот файл:



## Классы

- class [ServerClientInterface](#)  
Класс для взаимодействия сервера с клиентами

### 4.8.1 Подробное описание

Заголовочный файл для модуля [ServerClientInterface](#).

Автор

Попов Г.В.

Версия

10.0

Дата

13.12.2023

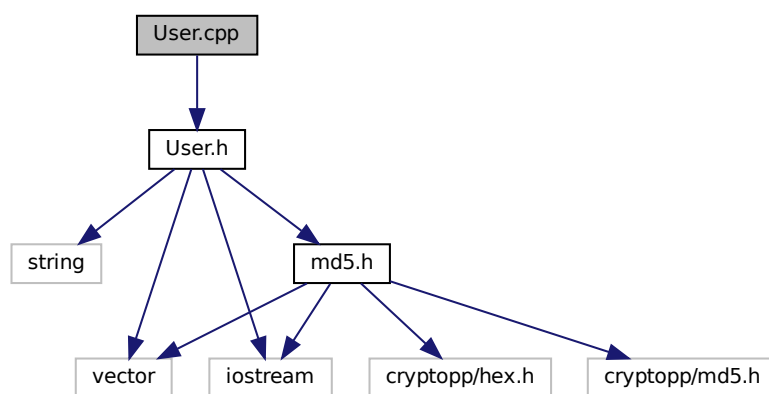
Авторство

ИБСТ ПГУ

## 4.9 Файл User.cpp

```
#include "User.h"
```

Граф включаемых заголовочных файлов для User.cpp:



## 4.10 Файл User.h

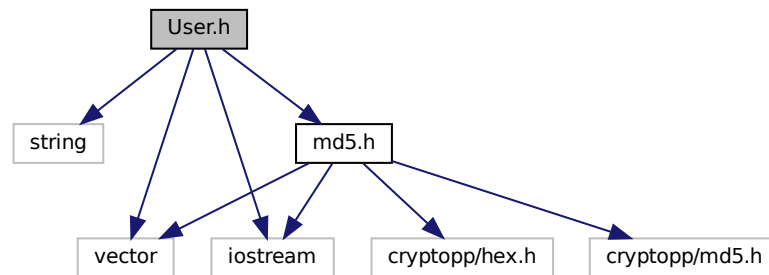
Заголовочный файл для модуля [User](#).

```
#include <string>
#include <vector>
#include <iostream>
```

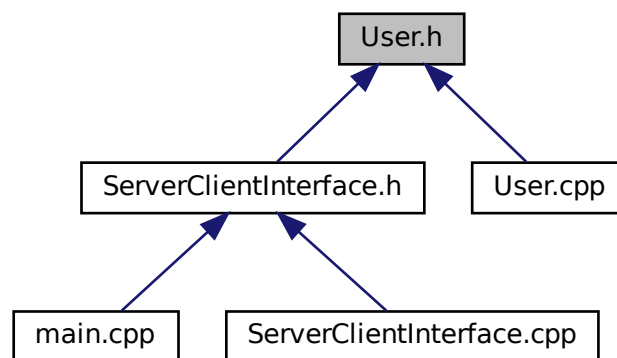


```
#include "md5.h"
```

Граф включаемых заголовочных файлов для User.h:



Граф файлов, в которые включается этот файл:



## Классы

- class `User`

Класс, представляющий информацию о подключенном пользователе

### 4.10.1 Подробное описание

Заголовочный файл для модуля `User`.

Автор

Попов Г.В.

Версия

1.0

Дата

13.12.2024

Авторство

ИБСТ ПГУ

# Предметный указатель

- CheckPassword
  - User, [11](#)
- DataReader, [5](#)
  - getClient, [5](#)
- DataReader.cpp, [13](#)
- DataReader.h, [13](#)
- DataReader\_fix, [6](#)
- error\_recording
  - Errors, [7](#)
- Errors, [6](#)
  - error\_recording, [7](#)
- Errors.cpp, [15](#)
- Errors.h, [15](#)
- Errors\_fix, [8](#)
- getClient
  - DataReader, [5](#)
- interaction
  - ServerClientInterface, [10](#)
- main.cpp, [17](#)
- md5.cpp, [17](#)
  - MD5\_hash, [18](#)
- MD5\_hash
  - md5.cpp, [18](#)
- Server\_fix, [8](#)
- ServerClientInterface, [9](#)
  - interaction, [10](#)
  - sum\_vec, [10](#)
- ServerClientInterface.cpp, [18](#)
- ServerClientInterface.h, [18](#)
- sum\_vec
  - ServerClientInterface, [10](#)
- User, [10](#)
  - CheckPassword, [11](#)
  - CheckLogin, [11](#)
- User.cpp, [20](#)
- User.h, [20](#)
- User\_fix, [12](#)
- CheckLogin
  - User, [11](#)