

Документация на использование программы сервера

Создано системой Doxygen 1.9.4

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>Result</code>	8
3.6	Структура <code>Server_fix</code>	9
3.7	Класс <code>ServerClientInterface</code>	9
3.7.1	Подробное описание	10
3.7.2	Методы	10
3.7.2.1	<code>interaction()</code>	10
3.7.2.2	<code>sum_vec()</code>	10
3.8	Класс <code>User</code>	11
3.8.1	Подробное описание	11
3.8.2	Методы	11
3.8.2.1	<code>CheckLogin()</code>	12
3.8.2.2	<code>CheckPassword()</code>	12
3.9	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	<code>DataReader.h</code>	15
4.4	Файл <code>Errors.cpp</code>	15
4.5	Файл <code>Errors.h</code>	16
4.5.1	Подробное описание	17
4.6	<code>Errors.h</code>	17
4.7	Файл <code>main.cpp</code>	17
4.7.1	Подробное описание	18
4.8	Файл <code>ServerClientInterface.cpp</code>	18
4.9	Файл <code>ServerClientInterface.h</code>	18

4.9.1 Подробное описание	19
4.10 ServerClientInterface.h	20
4.11 Файл User.cpp	21
4.12 Файл User.h	21
4.12.1 Подробное описание	22
4.13 User.h	23
Предметный указатель	25

Глава 1

Алфавитный указатель классов

1.1 Классы

Классы с их кратким описанием.

DataReader	
Класс для получения БД клиентов	5
DataReader_fix	6
Errors	
Класс обработки ошибок	6
Errors_fix	8
Result	8
Server_fix	9
ServerClientInterface	
Класс для взаимодействия сервера с клиентами	9
User	
Класс, представляющий информацию о подключенном пользователе	11
User_fix	12

Глава 2

Список файлов

2.1 Файлы

Полный список документированных файлов.

DataReader.cpp	13
DataReader.h	
Заголовочный файл для модуля DataReader	13
Errors.cpp	15
Errors.h	
Заголовочный файл для модуля Errors	16
main.cpp	
Функция для предоставления пользователю справки по использованию программы	17
ServerClientInterface.cpp	18
ServerClientInterface.h	
Заголовочный файл для модуля ServerClientInterface	18
User.cpp	21
User.h	
Заголовочный файл для модуля User	21

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 Errors 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

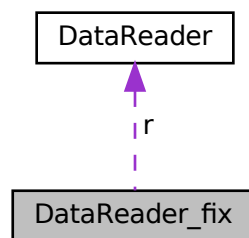
Errors	при возникновении ошибки
------------------------	--------------------------

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

<code>File_Log</code>	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

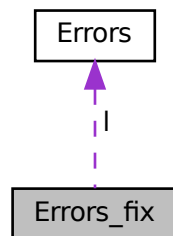
<code>flag</code>	Хранит в себе тип ошибки
<code>info</code>	Хранит в себе подробную информацию об ошибке

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 Result

Открытые атрибуты

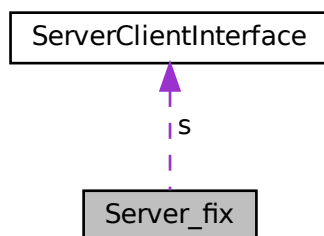
- `std::string hash`
- `std::string salt`
- `std::string login`

Объявления и описания членов структуры находятся в файле:

- [ServerClientInterface.cpp](#)

3.6 Структура Server_fix

Граф связей класса Server_fix:



Открытые атрибуты

- `ServerClientInterface * s`

Объявления и описания членов структуры находятся в файле:

- `unit.cpp`

3.7 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
- `uint64_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< uint64_t > get_vec ()`

- Сеттер для атрибута vec.
- `void set_vec (vector< uint64_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.

3.7.1 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 Errors обработки ошибок

3.7.2 Методы

3.7.2.1 interaction()

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

Главная функция для взаимодействия сервера с клиентами

Исключения

Errors	при возникновении ошибки
------------------------	--------------------------

3.7.2.2 sum_vec()

```
uint64_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](#)

3.8 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_password, vector< string > Db_ID, string SALT, string SendHash)`
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.

3.8.1 Подробное описание

Класс, представляющий информацию о подключенном пользователе

Аргументы

ID	ID подключенного клиента
hash	хэш-код подключенного клиента

3.8.2 Методы

3.8.2.1 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

3.8.2.2 CheckPassword()

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

Функция для проверки пароля подключенного клиента

Возвращает

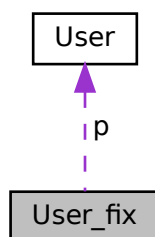
Возвращает true - если пароль подключенного клиента совпадает с паролем из БД, иначе false

Объявления и описания членов классов находятся в файлах:

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

3.9 Структура User_fix

Граф связей класса User_fix:



Открытые атрибуты

- [User * p](#)

Объявления и описания членов структуры находятся в файле:

- [unit.cpp](#)

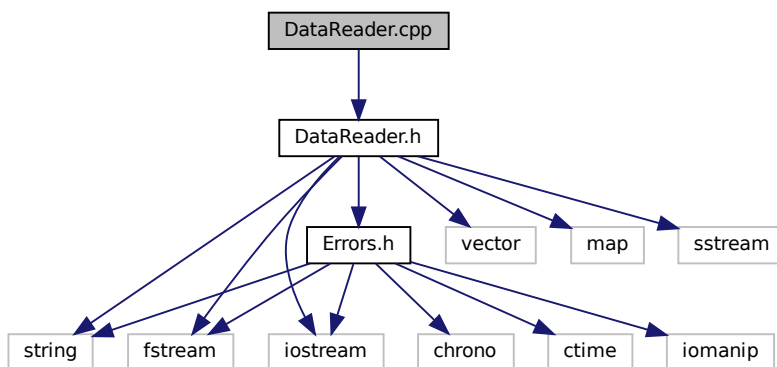
Глава 4

Файлы

4.1 Файл DataReader.cpp

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

Граф включаемых заголовочных файлов для DataReader.cpp:



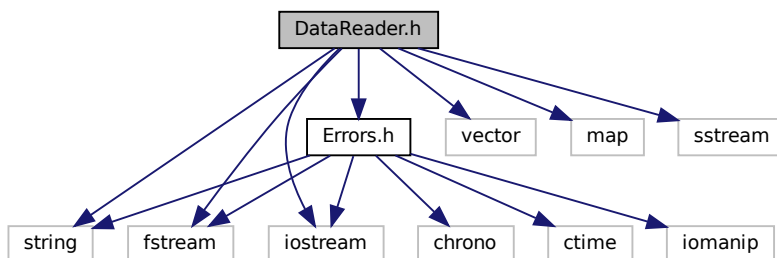
4.2 Файл DataReader.h

Заголовочный файл для модуля [DataReader](#).

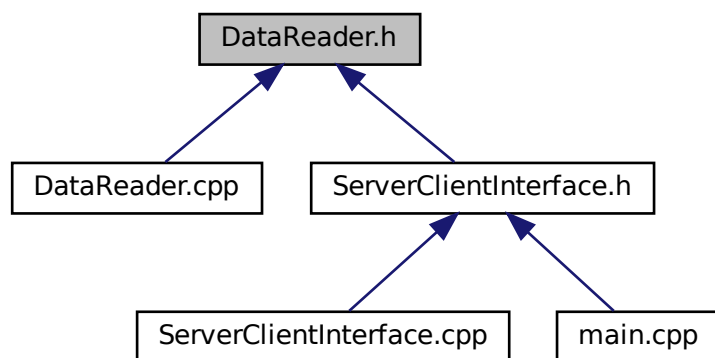
```
#include <string>
#include <vector>
#include <map>
#include <fstream>
#include <sstream>
#include <iostream>
```

```
#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

Маматкулова Р.А.

Версия

1.0

Дата

21.11.2024

Авторство

ИБСТ ПГУ

4.3 DataReader.h

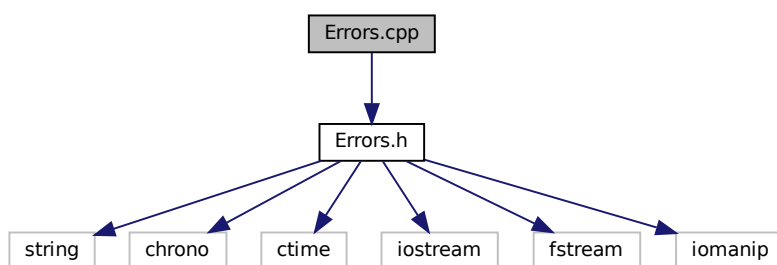
[См. документацию.](#)

```
1
10 #include <string>
11 #include <vector>
12 #include <map>
13 #include <fstream>
14 #include <sstream>
15 #include <iostream>
16
17 #include "Errors.h"
18
19 using namespace std;
20
21 class DataReader{
22     public:
23
24         string get_FileReader();
25         void set_FileReader(string file);
26
27         pair<vector<string>, vector<string>> getClient();
28     private:
29         string FileReader;
30         Errors Err;
31 };
32
```

4.4 Файл Errors.cpp

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

Граф включаемых заголовочных файлов для Errors.cpp:

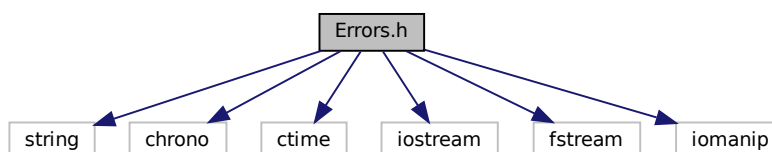


4.5 Файл Errors.h

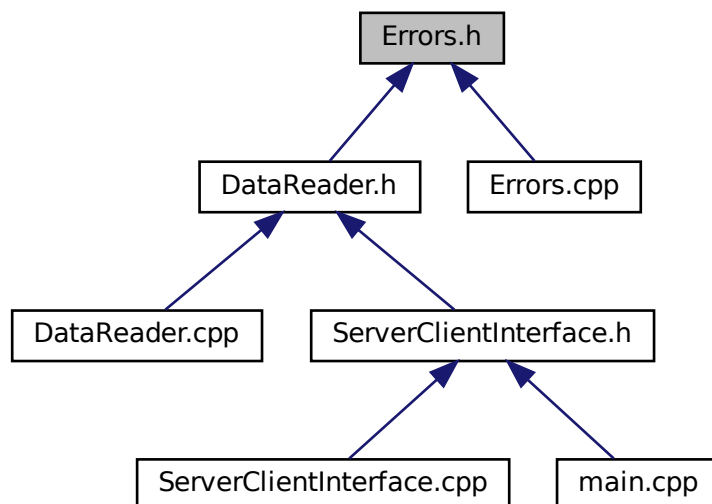
Заголовочный файл для модуля [Errors](#).

```
#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

4.5.1 Подробное описание

Заголовочный файл для модуля [Errors](#).

Автор

Маматкулова Р.А.

Версия

1.0

Дата

21.11.2024

Авторство

ИБСТ ПГУ

4.6 Errors.h

[См. документацию.](#)

```

1
10 #include <string>
11 #include <chrono>
12 #include <ctime>
13 #include <iostream>
14 #include <fstream>
15 #include <iomanip>
16
17 using namespace std;
18
23 class Errors{
24     public:
25         string get_File_Log();
26         void set_File_Log(string file);
27
28         void error_recording(string flag, string info);
29
30     private:
31         string File_Log;
32 };

```

4.7 Файл main.cpp

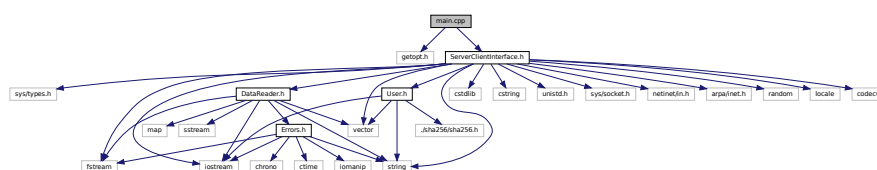
Функция для предоставления пользователю справки по использованию программы

```

#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

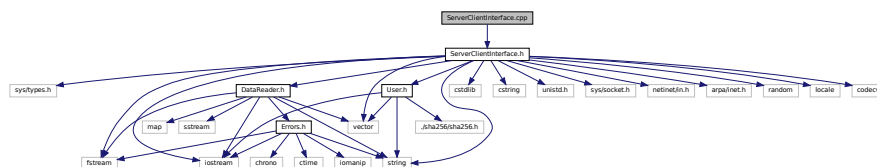
4.7.1 Подробное описание

Функция для предоставления пользователю справки по использованию программы

4.8 Файл ServerClientInterface.cpp

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

Граф включаемых заголовочных файлов для ServerClientInterface.cpp:



Классы

- struct Result

Функции

- **Result** splitString (const std::string &input)

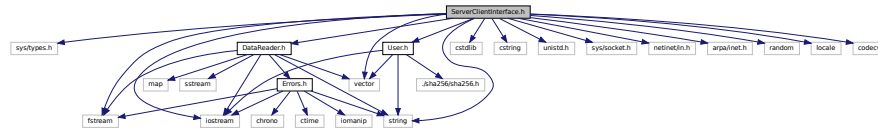
4.9 Файл ServerClientInterface.h

Заголовочный файл для модуля `ServerClientInterface`.

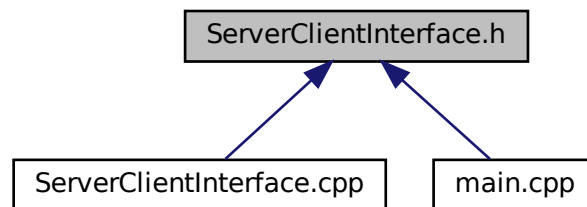
```
#include <sys/types.h>
#include <iostream>
#include <cstdlib>
#include <cstring>
#include <unistd.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <fstream>
#include <vector>
#include <random>
#include <string>
```

```
#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

4.9.1 Подробное описание

Заголовочный файл для модуля [ServerClientInterface](#).

Автор

Маматкулова Р.А.

Версия

1.0

Дата

21.11.2024

Авторство

ИБСТ ПГУ

4.10 ServerClientInterface.h

См. документацию.

```

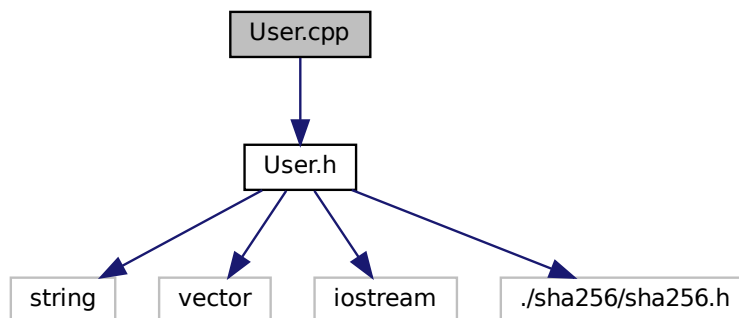
1
10 #pragma once
11 #include <sys/types.h>
12 #include <iostream>
13 #include <cstdlib>
14 #include <cstring>
15 #include <unistd.h>
16 #include <sys/socket.h>
17 #include <netinet/in.h>
18 #include <arpa/inet.h>
19 #include <fstream>
20 #include <vector>
21 #include <random>
22
23 #include <string>
24 #include <locale>
25 #include <codecvt>
26
27 #include "DataReader.h"
28 #include "User.h"
29
30 using namespace std;
31
32 class ServerClientInterface{
33     public:
34         int interaction(string database, string logFile);
35         uint64_t sum_vec();
36
37         string get_address();
38         void set_address(string address1);
39
40         int get_port();
41         void set_port(int port1);
42
43         vector<uint64_t> get_vec();
44         void set_vec(vector<uint64_t> v);
45
46         string get_salt();
47         void set_salt();
48
49         pair<vector<string>, vector<string>> get_DB_clients();
50         void set_DB_clients(vector<string> login, vector<string> password);
51
52     private:
53         string address;
54         int port;
55         vector<uint64_t> vec;
56         string salt;
57         pair<vector<string>, vector<string>>DB_clients;
58         Errors Err;
59 };

```


4.11 Файл User.cpp

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

Граф включаемых заголовочных файлов для User.cpp:



4.12 Файл User.h

Заголовочный файл для модуля [User](#).

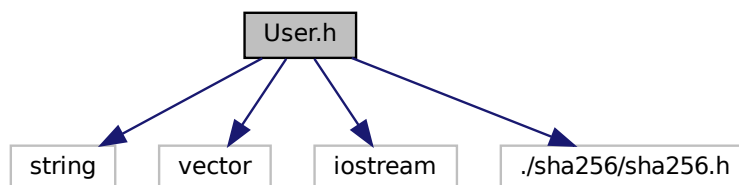
```
#include <string>
```

```
#include <vector>
```

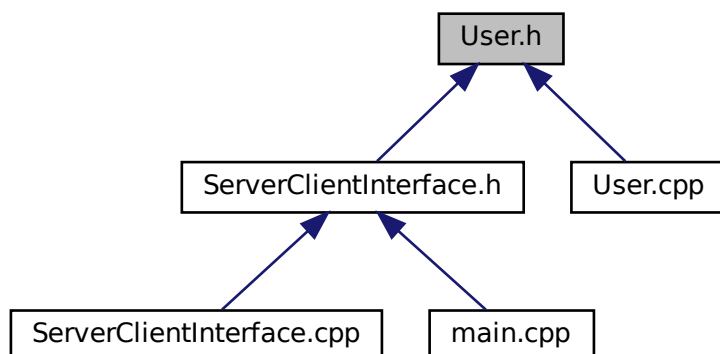
```
#include <iostream>
```

```
#include "./sha256/sha256.h"
```

Граф включаемых заголовочных файлов для User.h:



Граф файлов, в которые включается этот файл:



Классы

- class [User](#)

Класс, представляющий информацию о подключенном пользователе

4.12.1 Подробное описание

Заголовочный файл для модуля [User](#).

Автор

Маматкулова Р.А.

Версия

1.0

Дата

21.11.2024

Авторство

ИБСТ ПГУ

4.13 User.h

[См. документацию.](#)

```
1
10 #include <string>
11 #include <vector>
12 #include <iostream>
13
14 #include "../sha256/sha256.h"
15
16 using namespace std;
17
23 class User{
24     private:
25         string ID;
26         string hash;
27     public:
28         bool CheckLogin(vector<string> Db_ID);
29         bool CheckPassword(vector<string> Db_password, vector<string> Db_ID, string SALT, string SendHash);
30
31         string get_ID();
32         void set_ID(string ID1);
33
34         string get_hash();
35         void set_hash(string hash1);
36
37 };
```


Предметный указатель

- CheckLogin
 - User, [11](#)
- CheckPassword
 - User, [12](#)
- 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, [16](#)
- Errors_fix, [8](#)
- getClient
 - DataReader, [5](#)
- interaction
 - ServerClientInterface, [10](#)
- main.cpp, [17](#)
- Result, [8](#)
- Server_fix, [9](#)
- ServerClientInterface, [9](#)
 - interaction, [10](#)
 - sum_vec, [10](#)
- ServerClientInterface.cpp, [18](#)
- ServerClientInterface.h, [18](#)
- sum_vec
 - ServerClientInterface, [10](#)
- User, [11](#)
 - CheckLogin, [11](#)
 - CheckPassword, [12](#)
- User.cpp, [21](#)
- User.h, [21](#)
- User_fix, [12](#)