

LockVox

0.1.1

Generated by Doxygen 1.9.1

1 README	1
1.1 LockVox Project	1
1.1.1 Introduction	1
1.1.2 Current Status	1
1.1.2.1 Server side	1
1.1.2.2 Client side	2
1.1.2.3 TO DO	2
1.1.3 3rd-party	2
1.1.4 Authors	2
1.1.5 Contact	2
2 Bug List	3
3 Hierarchical Index	5
3.1 Class Hierarchy	5
4 Class Index	7
4.1 Class List	7
5 File Index	9
5.1 File List	9
6 Class Documentation	11
6.1 AbstractServer Class Reference	11
6.1.1 Detailed Description	12
6.2 CChannel Class Reference	13
6.2.1 Detailed Description	14
6.2.2 Constructor & Destructor Documentation	14
6.2.2.1 CChannel() [1/5]	14
6.2.2.2 CChannel() [2/5]	14
6.2.2.3 CChannel() [3/5]	15
6.2.2.4 CChannel() [4/5]	15
6.2.2.5 CChannel() [5/5]	16
6.2.3 Member Function Documentation	16
6.2.3.1 addUser()	16
6.2.3.2 delUser()	16
6.2.3.3 deserialize()	17
6.2.3.4 get_clients()	17
6.2.3.5 get_id()	17
6.2.3.6 get_maxUsers()	18
6.2.3.7 get_name()	18
6.2.3.8 get_nbClients()	18
6.2.3.9 getMessagesLists()	19
6.2.3.10 serializeToObj()	19

6.2.3.11 set_all()	19
6.2.3.12 set_clients()	20
6.2.3.13 set_id()	20
6.2.3.14 set_maxUsers()	20
6.2.3.15 set_name()	21
6.2.3.16 set_nbClients()	21
6.2.3.17 setMessagesLists()	21
6.3 CClient Class Reference	22
6.3.1 Detailed Description	24
6.3.2 Constructor & Destructor Documentation	25
6.3.2.1 CClient() [1/4]	26
6.3.2.2 CClient() [2/4]	26
6.3.2.3 CClient() [3/4]	26
6.3.2.4 CClient() [4/4]	27
6.3.2.5 ~CClient()	27
6.3.3 Member Function Documentation	27
6.3.3.1 createChannelIndex()	27
6.3.3.2 createMessageList()	28
6.3.3.3 deserialize()	28
6.3.3.4 get_description()	29
6.3.3.5 get_idChannel()	29
6.3.3.6 get_isAuthenticate()	29
6.3.3.7 get_isOnline()	30
6.3.3.8 get_mail()	30
6.3.3.9 get_profilePic()	30
6.3.3.10 get_pseudo()	31
6.3.3.11 get_socket()	31
6.3.3.12 get_uuid()	31
6.3.3.13 insertChannelIndex()	32
6.3.3.14 processData()	32
6.3.3.15 readChannelIndex()	32
6.3.3.16 serializeToObj()	33
6.3.3.17 set_all()	33
6.3.3.18 set_description()	33
6.3.3.19 set_idChannel()	34
6.3.3.20 set_isAuthenticate()	34
6.3.3.21 set_isOnline()	35
6.3.3.22 set_mail()	35
6.3.3.23 set_profilePic()	35
6.3.3.24 set_pseudo()	36
6.3.3.25 set_socket()	36
6.3.3.26 set_uuid()	36

6.4 CDatabase Class Reference	37
6.4.1 Detailed Description	37
6.5 ChannelList Class Reference	38
6.6 ChannelModel Class Reference	39
6.7 ClientList Class Reference	40
6.8 ClientModel Class Reference	41
6.9 CMessage Class Reference	42
6.9.1 Detailed Description	42
6.10 CookiesManager Class Reference	43
6.10.1 Detailed Description	43
6.11 CPacket Class Reference	44
6.11.1 Detailed Description	46
6.12 crole Class Reference	47
6.13 CServer Class Reference	47
6.13.1 Detailed Description	49
6.13.2 Constructor & Destructor Documentation	50
6.13.2.1 CServer()	50
6.14 CSessionCookie Class Reference	50
6.14.1 Detailed Description	50
6.15 ImageProvider Class Reference	51
6.16 MessageList Class Reference	52
6.17 MessageModel Class Reference	52
6.18 QSslServer Class Reference	53
6.18.1 Detailed Description	54
6.19 register_info Struct Reference	54
6.20 SHA256 Class Reference	55
6.20.1 Detailed Description	55
6.20.2 Member Data Documentation	56
6.20.2.1 sha256_k	56
6.21 UIWorker Class Reference	56
7 File Documentation	59
7.1 LockVox/Client/cclient.cpp File Reference	59
7.1.1 Detailed Description	59
7.2 LockVox/Server/cclient.cpp File Reference	59
7.2.1 Detailed Description	60
7.3 LockVox/Client/includes/cclient.h File Reference	60
7.3.1 Detailed Description	60
7.4 LockVox/Server/includes/cclient.h File Reference	61
7.4.1 Detailed Description	61
7.5 LockVox/Server/includes/cpacket.h File Reference	62
7.5.1 Detailed Description	62

7.6 LockVox/Server/includes/cserver.h File Reference	62
7.6.1 Detailed Description	63
7.7 LockVox/Server/includes/cdatabase.h File Reference	63
7.7.1 Detailed Description	64
7.8 LockVox/Server/includes/cookiesmanager.h File Reference	64
7.8.1 Detailed Description	64
7.9 LockVox/Server/includes/csessioncookie.h File Reference	65
7.9.1 Detailed Description	65
7.10 LockVox/Server/includes/qsslserver.h File Reference	65
7.10.1 Detailed Description	66
7.11 LockVox/Server/includes/sha256.h File Reference	66
7.11.1 Detailed Description	66
7.11.2 Macro Definition Documentation	67
7.11.2.1 SHA2_PACK32	67
7.11.2.2 SHA2_UNPACK32	67
7.12 LockVox/src/includes/AbstractServer.h File Reference	67
7.12.1 Detailed Description	68
7.13 LockVox/src/includes/cchannel.h File Reference	68
7.13.1 Detailed Description	68
7.14 LockVox/src/includes/cmessage.h File Reference	69
7.14.1 Detailed Description	69
Index	71

Chapter 1

README

1.1 LockVox Project

Lockvox project aims to create a fully encrypted and open source voice and audio exchange system while keeping the GUI as simple and easy to access as possible.

1.1.1 Introduction

We are a group of friends and classmate currently in IT engineering school. This project is our annual project as part of our 4th year of university course. It's dev using Qt opensource, written in C++.

[Documentation](#)

1.1.2 Current Status

The project actually have :

1.1.2.1 Server side

- A working server running on port 50885
- Connect to MariaDB / MySQL server
- Login
- Register
- Generating UUID for new user
- Password stored using sha256 (salt NYI)
- Channel list
- User list and isOnline info
- Simple public message
- Profile picture

1.1.2.2 Client side

- Simple GUI
- Connect to server
- Login
- Register
- Retrieve channel and user list
- Simple public message
- Profile pictures

1.1.2.3 TO DO

- [Audio implementation](#)
- Message picture implementation
- Emoji message implementation

1.1.3 3rd-party

- [Skycoder42/QSslServer](#)
- [EMIPLIB](#)
- [MariaDB C & C++ Connector](#)
- [Zedwood C++ SHA256](#)
- [Qt Open Sources](#)

1.1.4 Authors

- Valentin MONNOT
- Hugues MATTEI
- Hugo LAVEAU
- Louis BOURGEAIS
- François LELIEVRE

1.1.5 Contact

- Dev Team (dev@lockvox.fr)
- Admin (admin@lockvox.fr)

Chapter 2

Bug List

Class [AbstractServer](#)

Class [CChannel](#)

Class [CClient](#)

QAbstractSocket tells us that waitForBytes written isn't allowed in unconnected state when client alt-f4. Mybe it comes from onDisconnect from [CServer](#).

Class [CDatabase](#)

In db configuration file, if MY_HOSTNAME=domainName.something MariaDB connector didn't recognize it as an address and use your ip address, i don't f***ing know why. All work great if MY_HOSTNAME is hardcoded or if you use a standard IPV4 address in configuration file

Class [CMessage](#)

Class [CookiesManager](#)

Class [CPacket](#)

Class [CServer](#)

Class [CSessionCookie](#)

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

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

CChannel	13
CMessage	42
CookiesManager	43
CPacket	44
crole	47
CSessionCookie	50
QAbstractListModel	
ChannelModel	39
ClientModel	41
MessageModel	52
QObject	
AbstractServer	11
CServer	47
CServer	47
ChannelList	38
ClientList	40
MessageList	52
UIWorker	56
QQuickImageProvider	
ImageProvider	51
QTcpServer	
QSslServer	53
QThread	
CClient	22
CDatabase	37
register_info	54
SHA256	55

Chapter 4

Class Index

4.1 Class List

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

AbstractServer	
Declaration of the class AbstractServer	11
CChannel	
Declaration of the class CChannel	13
CClient	
Declaration of the class CClient	22
CDatabase	
SQL Database Class	37
ChannelList	38
ChannelModel	39
ClientList	40
ClientModel	41
CMessage	
Declaration of the class CMessage	42
CookiesManager	
Declaration of the class CookiesManager	43
CPacket	
Declaration of the class Server CPacket	44
crole	47
CServer	
Declaration of the class CServer	47
CSessionCookie	
Declaration of the class CSessionCookie	50
ImageProvider	51
MessageList	52
MessageModel	52
QSslServer	
An extension of QTcpServer to support ssl. The counterpart to QSslSocket	53
register_info	54
SHA256	
Sha256 C implementation	55
UIWorker	56

Chapter 5

File Index

5.1 File List

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

LockVox/ imageprovider.h	??
LockVox/Client/ cclient.cpp	
Definition of class CClient	59
LockVox/Client/includes/ caudio.h	??
LockVox/Client/includes/ cclient.h	
Declaration of the class CClient	60
LockVox/Client/includes/ channellist.h	??
LockVox/Client/includes/ channelmodel.h	??
LockVox/Client/includes/ clientlist.h	??
LockVox/Client/includes/ clientmodel.h	??
LockVox/Client/includes/ config.h	??
LockVox/Client/includes/ cpacket.h	??
LockVox/Client/includes/ cserver.h	??
LockVox/Client/includes/ messagelist.h	??
LockVox/Client/includes/ messagemodel.h	??
LockVox/Client/includes/ uiworker.h	??
LockVox/Server/ cclient.cpp	
Definition of class CClient	59
LockVox/Server/ config.h	??
LockVox/Server/includes/ caudio.h	??
LockVox/Server/includes/ cclient.h	
Declaration of the class CClient	61
LockVox/Server/includes/ cdatabase.h	
Declaration of the class CDatabase	63
LockVox/Server/includes/ cookiesmanager.h	
Declaration of the class CookieManager	64
LockVox/Server/includes/ cpacket.h	
Declaration of the class CServer CPacket	62
LockVox/Server/includes/ cserver.h	
Declaration of the class CServer	62
LockVox/Server/includes/ csessioncookie.h	
Declaration of the class CSessionCookie	65
LockVox/Server/includes/ qsslserver.h	
An extension of QTcpServer to support ssl. The counterpart to QSslSocket	65
LockVox/Server/includes/ sha256.h	
Sha256 C implementation	66

LockVox/src/includes/ AbstractServer.h	
Declaration of the class AbstractServer	67
LockVox/src/includes/ cchannel.h	
Declaration of the class CChannel	68
LockVox/src/includes/ cmessage.h	
Declaration of the class CMessage	69
LockVox/src/includes/ crole.h	??

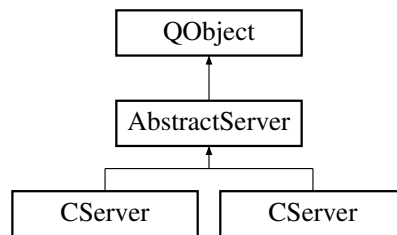
Chapter 6

Class Documentation

6.1 AbstractServer Class Reference

Declaration of the class [AbstractServer](#).

Inheritance diagram for AbstractServer:



Public Member Functions

- `QList< CChannel * > get_channelList ()`
- `QList< CClient * > get_clientList ()`
- `QByteArray * get_clientBuffer (int id)`
- `QList< CChannel > get_staticChannelList ()`
- `QList< CClient > get_staticClientList ()`
- `CChannel * get_channelById (int id)`
- `CClient * get_clientById (QUuid uuid)`
- `void set_clients (QList< CClient * > clients)`
- `void set_channels (QList< CChannel * > channels)`
- `void set_channel (CChannel channel, int index)`
- `void freeChannels ()`
- `void freeClients ()`
- `void addClient (CClient *client)`
- `void addChannel (CChannel *channel)`
- `void delClient (CClient *client)`
- `void DelChannel (CChannel *channel)`
- `void delClient (int index)`
- `void DelChannel (int index)`
- `void printChannels ()`
- `void printClients ()`

Protected Attributes

- QList< [CChannel](#) * > **m_channels**
- QList< [CClient](#) * > **m_clients**
- QList< QByteArray * > **m_buffers**

6.1.1 Detailed Description

Declaration of the class [AbstractServer](#).

This class represent the virtual class of [CServer](#)

Author

LockVox Team

Version

0.2.0

Date

2021

Note

Bug

Warning

Remarks

Copyright

GNU GPL-3.0

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

- LockVox/src/includes/[AbstractServer.h](#)
- LockVox/src/AbstractServer.cpp

6.2 CChannel Class Reference

Declaration of the class [CChannel](#).

Public Member Functions

- [CChannel](#) ()
Default constructor of the class [CChannel](#).
- [CChannel](#) (QString name, int id)
Custom Constructor of the class [CChannel](#) This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.
- [CChannel](#) (QString name, int id, int maxUsers)
Custom Constructor of the class [CChannel](#) This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.
- [CChannel](#) (QList< [CClient](#) * > clients, QString name, int id)
Custom Constructor of the class [CChannel](#) This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.
- [CChannel](#) (const [CChannel](#) ©)
Copy constructor of the class [CChannel](#) This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.
- QList< [CClient](#) * > [get_clients](#) ()
Getter of attribute m_clients.
- QString [get_name](#) ()
Getter of attribute m_name.
- [CChannel](#) * [get_this](#) ()
- int [get_id](#) ()
Getter of attribute m_id.
- int [get_maxUsers](#) ()
Getter of attribute m_maxUsers.
- int [get_nbClients](#) ()
Getter of attribute m_nbClients.
- MessageList * [getMessagesLists](#) () const
Getter of attribute m_messagesLists.
- void [set_clients](#) (QList< [CClient](#) * > clients)
Setter of attribute m_clients.
- void [set_name](#) (QString name)
Setter of attribute m_name.
- void [set_id](#) (int id)
Setter of attribute m_id.
- void [set_maxUsers](#) (int maxUsers)
Setter of attribute m_maxUsers.
- void [set_nbClients](#) (int nb)
Setter of attribute m_nbClients.
- void [set_all](#) ([CChannel](#) *c)
Set the different attribute of a [CChannel](#) from another an existing instance of [CChannel](#).
- void [setMessagesLists](#) (MessageList *messagesLists)
Setter of attribute m_messagesLists.
- QJsonObject [serializeToObj](#) ()
Serialize attributes of an user into a JSON object.
- void [deserialize](#) (QJsonObject json_obj)
Deserialize a JSON object into the current instance of [CChannel](#).
- void [addUser](#) ([CClient](#) *c)
Add an user to the list of users of this [CChannel](#) instance.
- void [delUser](#) (QUuid idUser)
Remove an user to the list of users of this [CChannel](#) instance.

6.2.1 Detailed Description

Declaration of the class [CChannel](#).

This class represent a channel

Author

LockVox Team

Version

0.1.1

Date

2021

Note

Bug

Warning

Remarks

Copyright

GNU GPL-3.0

6.2.2 Constructor & Destructor Documentation

6.2.2.1 CChannel() [1/5]

```
CChannel::CChannel ( )
```

Default constructor of the class [CChannel](#).

See also

[CChannel::CChannel\(\)](#)

6.2.2.2 CChannel() [2/5]

```
CChannel::CChannel (
    QString name,
    int id )
```

Custom Constructor of the class [CChannel](#) This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Parameters

<i>name</i>	the name of the channel
<i>id</i>	the ID of the channel

See also

[CChannel::CChannel\(\)](#)

6.2.2.3 CChannel() [3/5]

```
CChannel::CChannel (
    QString name,
    int id,
    int maxUsers )
```

Custom Constructor of the class [CChannel](#) This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Parameters

<i>id</i>	the ID of the channel
<i>name</i>	the name of the channel
<i>maxUsers</i>	the maximum number of user in the channel

See also

[CChannel::CChannel\(\)](#)

6.2.2.4 CChannel() [4/5]

```
CChannel::CChannel (
    QList< CClient * > clients,
    QString name,
    int id )
```

Custom Constructor of the class [CChannel](#) This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Parameters

<i>name</i>	the name of the channel
<i>id</i>	the ID of the channel
<i>clients</i>	the lists of clients who are in the channel

See also

[CChannel::CChannel\(\)](#)

6.2.2.5 CChannel() [5/5]

```
CChannel::CChannel (
    const CChannel & copy )
```

Copy constructor of the class [CChannel](#) This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Parameters

<i>copy</i>	the attribut to copy
-------------	----------------------

See also

[CChannel::CChannel\(\)](#)

6.2.3 Member Function Documentation

6.2.3.1 addUser()

```
void CChannel::addUser (
    CClient * c )
```

Add an user to the list of users of this [CChannel](#) instance.

Parameters

<i>in</i>	<i>c</i>	the user to add to the list of users
-----------	----------	--------------------------------------

6.2.3.2 delUser()

```
void CChannel::delUser (
    QUuid idUser )
```

Remove an user to the list of users of this [CChannel](#) instance.

Parameters

in	<i>idUser</i>	the ID of the user to remove to the list of users
----	---------------	---

6.2.3.3 deserialize()

```
void CChannel::deserialize (
    QObject json_obj )
```

Deserialize a JSON object into the current instance of [CChannel](#).

Parameters

in	<i>json_obj</i>	the JSON object to deserialize into this CChannel instance.
----	-----------------	---

6.2.3.4 get_clients()

```
QList< CClient * > CChannel::get_clients ( )
```

Getter of attribute `m_clients`.

Returns

`m_clients` the value of attribute `m_clients`

Return values

<i>QList<CClient</i>	<i>*></i>	the value of attribute <code>m_clients</code>
-------------------------	--------------	---

6.2.3.5 get_id()

```
int CChannel::get_id ( )
```

Getter of attribute `m_id`.

Returns

`m_id` the value of attribute `m_id`

Return values

<i>int</i>	the value of attribute m_id
------------	-----------------------------

6.2.3.6 get_maxUsers()

```
int CChannel::get_maxUsers ( )
```

Getter of attribute m_maxUsers.

Returns

m_maxUsers the value of attribute m_maxUsers

Return values

<i>int</i>	the value of attribute m_maxUsers
------------	-----------------------------------

6.2.3.7 get_name()

```
QString CChannel::get_name ( )
```

Getter of attribute m_name.

Returns

m_name the value of attribute m_name

Return values

<i>QString</i>	the value of attribute m_name
----------------	-------------------------------

6.2.3.8 get_nbClients()

```
int CChannel::get_nbClients ( )
```

Getter of attribute m_nbClients.

Returns

m_nbClients the value of attribute m_nbClients

Return values

<i>int</i>	the value of attribute <code>m_nbClients</code>
------------	---

6.2.3.9 getMessagesLists()

```
MessageList * CChannel::getMessagesLists ( ) const
```

Getter of attribute `m_messagesLists`.

Returns

`m_messagesLists` the value of attribute `m_messagesLists`

Return values

<i>MessageList</i>	the value of attribute <code>m_messagesLists</code>
--------------------	---

6.2.3.10 serializeToObj()

```
JsonObject CChannel::serializeToObj ( )
```

Serialize attributes of an user into a JSON object.

Parameters

out	<i>obj</i>	the JSON object which contains the attribute of the channel.
-----	------------	--

6.2.3.11 set_all()

```
void CChannel::set_all (
    CChannel * c )
```

Set the different attribute of a `CChannel` from another an existing instance of `CChannel`.

Parameters

in	<i>c</i>	the <code>CChannel</code> instance which contains the attribute to copy.
----	----------	--

6.2.3.12 set_clients()

```
void CChannel::set_clients (
    QList< CClient * > clients )
```

Setter of attribute m_clients.

Parameters

<i>clients</i>	the list of clients who are in the channel
----------------	--

Exceptions

<i>range_error</i>	if clients is empty
--------------------	---------------------

6.2.3.13 set_id()

```
void CChannel::set_id (
    int id )
```

Setter of attribute m_id.

Parameters

<i>id</i>	the ID of the channel
-----------	-----------------------

Exceptions

<i>range_error</i>	if ID is lower than 0
--------------------	-----------------------

6.2.3.14 set_maxUsers()

```
void CChannel::set_maxUsers (
    int maxUsers )
```

Setter of attribute m_maxUsers.

Parameters

<i>maxUsers</i>	the maximum number of users in the channel
-----------------	--

Exceptions

--	--

6.2.3.15 set_name()

```
void CChannel::set_name (
    QString name )
```

Setter of attribute m_name.

Parameters

<i>name</i>	the name of the channel
-------------	-------------------------

Exceptions

<i>range_error</i>	if name is empty
--------------------	------------------

6.2.3.16 set_nbClients()

```
void CChannel::set_nbClients (
    int nb )
```

Setter of attribute m_nbClients.

Parameters

<i>nb</i>	the number of users in the channel
-----------	------------------------------------

Exceptions

<i>range_error</i>	if the number of users is greater than the maximum number of users in the channel
--------------------	---

6.2.3.17 setMessagesLists()

```
void CChannel::setMessagesLists (
    MessageList * messagesList )
```

Setter of attribute m_messagesLists.

Parameters

<i>messagesList</i>	the list of messages associate to this channel
---------------------	--

Exceptions

--	--

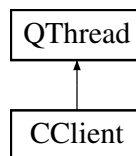
The documentation for this class was generated from the following files:

- LockVox/src/includes/cchannel.h
- LockVox/src/cchannel.cpp

6.3 CClient Class Reference

Declaration of the class [CClient](#).

Inheritance diagram for CClient:



Signals

- void **writeToLog** (QString error, int level)
- void **sendYou** (QTcpSocket *socket)
- void **updateMe** (int update_type, [CClient](#) *client)
- void **updateChan** (int update_type, [CChannel](#) *channel)
- void **authMe** (QList< QString > info, [CClient](#) *client)
- void **regMe** (QList< QString > info, [CClient](#) *Client)
- [CChannel](#) * **whichChan** (int id)
- void **sendToAll** (QByteArray out)

Public Member Functions

- [CClient](#) ()
Default constructor of the class [CClient](#).
- [CClient](#) (const [CClient](#) ©)
Copy constructor of the class [CClient](#) This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.
- [CClient](#) (QUuid id, QString name, QTcpSocket *soc, int idChannel, bool online, QString description)
Custom Constructor of the class [CClient](#) This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.
- [CClient](#) (QTcpSocket *soc)
Custom Constructor of the class [CClient](#) This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.
- [~CClient](#) ()
Default destructor of the class [CClient](#).
- QString [get_pseudo](#) ()

- Getter of attribute *m_pseudo*.
- QString [get_mail](#) ()
 - Getter of attribute *m_mail*.
- QString [get_description](#) ()
 - Getter of attribute *m_description*.
- QTcpSocket * [get_socket](#) ()
 - Getter of attribute *m_soc*.
- int [get_idChannel](#) ()
 - Getter of attribute *m_idChannel*.
- bool [get_isOnline](#) ()
 - Getter of attribute *m_isOnline*.
- bool [get_isAuthenticate](#) ()
 - Getter of attribute *m_isAuthenticate*.
- QUuid [get_uuid](#) ()
 - Getter of attribute *m_uuid*.
- QImage [get_profilePic](#) ()
 - Getter of attribute *profilePic*.
- void [set_pseudo](#) (QString pseudo)
 - Setter of attribute *m_pseudo*.
- void [set_mail](#) (QString mail)
 - Setter of attribute *m_mail*.
- void [set_socket](#) (QTcpSocket *soc)
 - Setter of attribute *m_soc*.
- void [set_idChannel](#) (int id)
 - Setter of attribute *m_idChannel*.
- void [set_all](#) (CClient *c)
 - Set the different attribute of a [CClient](#) from another an existing instance of [CClient](#).
- void [set_isOnline](#) (bool online)
 - Setter of attribute *m_isOnline*.
- void [set_isAuthenticate](#) (bool Auth)
 - Setter of attribute *m_isAuthenticate*.
- void [set_description](#) (QString d)
 - Setter of attribute *m_description*.
- void [set_uuid](#) (QUuid uuid)
 - Setter of attribute *m_uuid*.
- void [set_profilePic](#) (QImage img)
 - Setter of attribute *profilePic*.
- QJsonObject [serializeToObj](#) ()
 - Serialize attributes of an user into a JSON object.
- void [deserialize](#) (QJsonObject json_obj)
 - Deserialize a JSON object into the current instance of [CClient](#).
- **CClient** (const [CClient](#) ©)
- **CClient** (QUuid id, QString name, QTcpSocket *soc, int idChannel, bool online, QString description)
- **CClient** (QTcpSocket *soc)
- QString [get_pseudo](#) ()
- QString [get_mail](#) ()
- QString [get_description](#) ()
- QTcpSocket * [get_socket](#) ()
- int [get_idChannel](#) ()
- bool [get_isOnline](#) ()
- bool [get_isAuthenticate](#) ()

- QUuid **get_uuid** ()
- QImage **get_profilePic** ()
- void **set_pseudo** (QString pseudo)
- void **set_mail** (QString mail)
- void **set_socket** (QTcpSocket *soc)
- void **set_idChannel** (int id)
- void **set_all** (CClient *c)
- void **set_isOnline** (bool online)
- void **set_isAuthenticate** (bool Auth)
- void **set_description** (QString d)
- void **set_uuid** (QUuid uuid)
- void **set_profilePic** (QImage img)
- QJsonObject **serializeToObj** ()
- void **deserialize** (QJsonObject json_obj)
- void **processData** (QByteArray data)

This function is called by onReceiveData to process received data in order to manage cases defined by the application protocol.

- bool **createChannelIndex** (string filename, QString path_to_index)
Create an index.json for an empty channel and add the first message of it.
- QList< QString > **readChannelIndex** (QString path_to_index)
returns the list of names of all messages files contained in the given index
- bool **insertChannelIndex** (QString path_to_index, QList< QString > filename_list)
Update index.json when inserting new message to it.
- QList< CMessage > **createMessageList** (QString id, bool isPrivate, int nb_msg_to_sync, QUuid sender, int start_index)
Create a QList of CMessage stored locally using index.json.

6.3.1 Detailed Description

Declaration of the class [CClient](#).

This class represent an user

Author

Version

0.1.1

Date

2021

Note

[Bug](#)

Warning

Une mauvaise utilisation peut faire planter votre application (c'est votre faute)

Remarks

Une remarque à faire ?

Copyright

GNU GPL-3.0

This class represent an user

Author

LockVox Team

Version

0.2.0

Date

2021

Note

Bug QAbstractSocket tells us that waitForBytes written isn't allowed in unconnected state when client alt-f4. Mybe it comes from onDisconnect from [CServer](#).

Warning**Remarks****Copyright**

GNU GPL-3.0

6.3.2 Constructor & Destructor Documentation

6.3.2.1 CClient() [1/4]

```
CClient::CClient ( )
```

Default constructor of the class [CClient](#).

See also

[CClient::Client\(\)](#)

6.3.2.2 CClient() [2/4]

```
CClient::CClient (
    const CClient & copy )
```

Copy constructor of the class [CClient](#) This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Parameters

<i>copy</i>	the attribut to copy
-------------	----------------------

See also

[CClient::CClient\(\)](#)

6.3.2.3 CClient() [3/4]

```
CClient::CClient (
    QUuid id,
    QString pseudo,
    QTcpSocket * soc,
    int idChannel,
    bool online,
    QString description )
```

Custom Constructor of the class [CClient](#) This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Parameters

<i>id</i>	the UUID of the user
<i>pseudo</i>	the username of the user
<i>soc</i>	the socket used by the user
<i>idChannel</i>	the ID of the current channel
<i>online</i>	either the user is online or not
<i>description</i>	the description of the user

See also

[CClient::CClient\(\)](#)

6.3.2.4 CClient() [4/4]

```
CClient::CClient (
    QTcpSocket * soc )
```

Custom Constructor of the class [CClient](#) This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Parameters

<i>soc</i>	the socket used by the user
------------	-----------------------------

See also

[CClient::CClient\(\)](#)

6.3.2.5 ~CClient()

```
CClient::~~CClient ( )
```

Default desctructor of the class [CClient](#).

See also

[CClient::Client\(\)](#)

6.3.3 Member Function Documentation

6.3.3.1 createChannelIndex()

```
bool CClient::createChannelIndex (
    string filename,
    QString path_to_index )
```

Create an index.json for an empty channel and add the first message of it.

Parameters

in	<i>filename</i>	should be the exacte file name of the message file
in	<i>path_to_index</i>	should be something like : "storage/[public private]/[id]/index.json"

Returns

true if channel index has been created successfully, otherwise false

6.3.3.2 createMessageList()

```
QList< CMessage > CClient::createMessageList (
    QString id,
    bool isPrivate,
    int nb_msg_to_sync,
    QUuid sender,
    int start_index )
```

Create a QList of [CMessage](#) stored locally using index.json.

Parameters

in	<i>id</i>	refers to id of channel or user for storage
in	<i>isPrivate</i>	tells if it's a private message or not
in	<i>nb_msg_to_sync</i>	to tell how much message you want to sync should be -1 to retrieve all
in	<i>sender</i>	refere tu uuid of sender, used to gather private message
in	<i>start_index</i>	is used to retrieve older message, should be 0 if no message has already been sync

Returns

A list of message

Return values

A	Qlist of CMessage countaining all message selected according to given information, if an error has occured, return the error has name of the sender (field "from") in the first message of the QList
---	--

6.3.3.3 deserialize()

```
void CClient::deserialize (
    QJsonObject json_obj )
```

Deserialize a JSON object into the current instance of [CClient](#).

Parameters

<i>in</i>	<i>json_obj</i>	the JSON object to deserialize into this CClient instance.
-----------	-----------------	--

6.3.3.4 get_description()

```
QString CClient::get_description ( )
```

Getter of attribute m_description.

Returns

m_description the value of attribute m_description

Return values

<i>QString</i>	the value of attribute m_description
----------------	--------------------------------------

6.3.3.5 get_idChannel()

```
int CClient::get_idChannel ( )
```

Getter of attribute m_idChannel.

Returns

m_idChannel the value of attribute m_idChannel

Return values

<i>int</i>	the value of attribute m_idChannel
------------	------------------------------------

6.3.3.6 get_isAuthenticate()

```
bool CClient::get_isAuthenticate ( )
```

Getter of attribute m_isAuthenticate.

Returns

m_isAuthenticate the value of attribute m_isAuthenticate

Return values

<i>bool</i>	the value of attribute <code>m_isAuthenticate</code>
-------------	--

6.3.3.7 `get_isOnline()`

```
bool CClient::get_isOnline ( )
```

Getter of attribute `m_isOnline`.

Returns

`m_isOnline` the value of attribute `m_isOnline`

Return values

<i>bool</i>	the value of attribute <code>m_isOnline</code>
-------------	--

6.3.3.8 `get_mail()`

```
QString CClient::get_mail ( )
```

Getter of attribute `m_mail`.

Returns

`m_mail` the value of attribute `m_mail`

Return values

<i>QString</i>	the value of attribute <code>m_mail</code>
----------------	--

6.3.3.9 `get_profilePic()`

```
QImage CClient::get_profilePic ( )
```

Getter of attribute `profilePic`.

Returns

`profilePic` the value of attribute `profilePic`

Return values

<i>QImage</i>	the value of attribute profilePic
---------------	-----------------------------------

6.3.3.10 get_pseudo()

```
QString CClient::get_pseudo ( )
```

Getter of attribute m_pseudo.

Returns

m_pseudo the value of attribute m_pseudo

Return values

<i>QString</i>	the value of attribute m_pseudo
----------------	---------------------------------

6.3.3.11 get_socket()

```
QTcpSocket * CClient::get_socket ( )
```

Getter of attribute m_soc.

Returns

m_soc the value of attribute m_soc

Return values

<i>QTcpSocket</i>	the value of attribute m_soc
-------------------	------------------------------

6.3.3.12 get_uuid()

```
QUuid CClient::get_uuid ( )
```

Getter of attribute m_uuid.

Returns

m_uuid the value of attribute m_uuid

Return values

<i>QUuid</i>	the value of attribute m_uuid
--------------	-------------------------------

6.3.3.13 insertChannelIndex()

```
bool CClient::insertChannelIndex (
    QString path_to_index,
    QList< QString > filename_list )
```

Update index.json when inserting new message to it.

Parameters

in	<i>path_to_index</i>	should be something like : "storage/[public private]/[id]/index.json"
in	<i>filename_list</i>	is the list of all filename of message stored into for the given channel, can be extracted with CClient::readChannelIndex

Returns

true if the function had successfully updated the index, otherwise false

6.3.3.14 processData()

```
void CClient::processData (
    QByteArray data )
```

This function is called by onReceiveData to process received data in order to manage cases defined by the application protocol.

Parameters

in	<i>data</i>	The QByteArray object received from the TCP socket
----	-------------	--

6.3.3.15 readChannelIndex()

```
QList< QString > CClient::readChannelIndex (
    QString path_to_index )
```

returns the list of names of all messages files contained in the given index

Parameters

in	<i>path_to_index</i>	should be something like : "storage/[public private]/[id]/index.json"
----	----------------------	---

Returns

A list of message filename

Return values

A	QList of QString countaining all message filename in the given index. Return an empty QList if no index or empty index was found
---	--

6.3.3.16 serializeToObj()

```
QJsonObject CClient::serializeToObj ( )
```

Serialize attributes of an user into a JSON object.

Parameters

out	<i>obj</i>	the JSON object which contains the attribute of the user.
-----	------------	---

Returns

obj the JSON object which contains the attribute of the user.

6.3.3.17 set_all()

```
void CClient::set_all (
    CClient * c )
```

Set the different attribute of a [CClient](#) from another an existing instance of [CClient](#).

Parameters

in	<i>c</i>	the CClient instance which contains the attribute to copy.
----	----------	--

6.3.3.18 set_description()

```
void CClient::set_description (
```

```
QString d )
```

Setter of attribute m_description.

Parameters

<i>d</i>	the description of the user
----------	-----------------------------

Exceptions

--	--

6.3.3.19 set_idChannel()

```
void CClient::set_idChannel (
    int id )
```

Setter of attribute m_idChannel.

Parameters

<i>id</i>	the ID of the current channel
-----------	-------------------------------

Exceptions

<i>range_error</i>	if id is lower than -1
--------------------	------------------------

6.3.3.20 set_isAuthenticate()

```
void CClient::set_isAuthenticate (
    bool Auth )
```

Setter of attribute m_isAuthenticate.

Parameters

<i>Auth</i>	either the user has passed authentication or not
-------------	--

Exceptions

--	--

6.3.3.21 set_isOnline()

```
void CClient::set_isOnline (
    bool online )
```

Setter of attribute m_isOnline.

Parameters

<i>online</i>	either the user is online or not
---------------	----------------------------------

Exceptions

--	--

6.3.3.22 set_mail()

```
void CClient::set_mail (
    QString mail )
```

Setter of attribute m_mail.

Parameters

<i>mail</i>	the email of the user
-------------	-----------------------

Exceptions

<i>range_error</i>	if mail is empty
--------------------	------------------

6.3.3.23 set_profilePic()

```
void CClient::set_profilePic (
    QImage img )
```

Setter of attribute profilePic.

Parameters

<i>img</i>	the profile picture of the user
------------	---------------------------------

Exceptions

<i>range_error</i>	if img is null
--------------------	----------------

6.3.3.24 set_pseudo()

```
void CClient::set_pseudo (
    QString pseudo )
```

Setter of attribute m_pseudo.

Parameters

<i>pseudo</i>	the username of the user
---------------	--------------------------

Exceptions

<i>range_error</i>	if pseudo is empty
--------------------	--------------------

6.3.3.25 set_socket()

```
void CClient::set_socket (
    QTcpSocket * soc )
```

Setter of attribute m_soc.

Parameters

<i>soc</i>	the socket of the user
------------	------------------------

Exceptions

--	--

6.3.3.26 set_uuid()

```
void CClient::set_uuid (
    QUuid uuid )
```

Setter of attribute m_uuid.

Parameters

<i>uuid</i>	the UUID of the user
-------------	----------------------

Exceptions

<code>range_error</code>	if uuid is null
--------------------------	-----------------

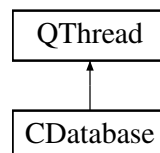
The documentation for this class was generated from the following files:

- LockVox/Client/includes/[cclient.h](#)
- LockVox/Client/[cclient.cpp](#)

6.4 CDatabase Class Reference

SQL Database Class.

Inheritance diagram for CDatabase:



Public Member Functions

- QString **init** ()
- string **getHash** (string id, QString *get_err)
- QString **newUser** (string uuid, string pseudo, string mail, string password)
- QString **updateUser** (string uuid, string pseudo, string mail, string description)
- QString **updateDescription** (string uuid, string description)
- QString **updateUsername** (string uuid, string username)
- QString **changePassword** (string mail, string uuid, string password, string newPass)
- QString **updateChannel** (string id, string name, string maxuser)
- QString **deleteUser** (string uuid)
- QString **deleteChannel** (string id)
- QList< [CChannel](#) * > **parseChannel** ()
- QList< [CClient](#) * > **parseClient** ()
- [CClient](#) * **parseClient** (string email, QString *get_err)

6.4.1 Detailed Description

SQL Database Class.

This class represents the main database (MySQL/MariaDB) and its function using mariadb connector lib

Author

Valentin Monnot (LockVox Team)

Version

0.2.0

Date

2021

Note

db.cnf in "%LOCKVOX_SERVER_ROOT_DIR%/storage/conf/" use 4 parameter, all of them should be there even if not used !

Order is important ! If you didn't use one of the parameter, default value will be applied, except for MY_PASSWORD. For user security you should never use a database without password...

Exemple with default values :

MY_HOSTNAME=127.0.0.1

MY_DATABASE=lockvox_server

MY_USERNAME=lockvox

MY_PASSWORD=changeme

If no configuration file found when starting the server, it will automaticcaly create a db.cnf file with these default values. After what server will close telling you that its default values and you should change them.

Bug In db configuration file, if MY_HOSTNAME=domainName.something MariaDB connector didn't recognize it as an address and use your ip address, i don't f***ing know why. All work great if MY_HOSTNAME is hardcoded or if you use a standard IPV4 address in configuration file

Warning

DEV_DB is our MariaDB database used for testing purpose, only fake account with troll username/password should be used on this database as everyone with an internet connection could access it ! if you just want to test, use account "test" with password "test". If you want to use your own MariaDB/MySQL database, we advise you to NOT hardcode your database credential. You should go in "config.h" and comment #define DEV_DB so you can use db.cnf file to configure your own DB.

Remarks**Copyright**

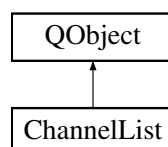
GNU GPL-3.0

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

- LockVox/Server/includes/[cdatabase.h](#)
- LockVox/src/cdatabase.cpp

6.5 ChannelList Class Reference

Inheritance diagram for ChannelList:



Public Slots

- void **appendItem** ()
- void **removeItem** (int index)

Signals

- void **dataChanged** ()
- void **preItemAppended** ()
- void **postItemAppended** ()
- void **preItemRemoved** (int index)
- void **postItemRemoved** ()

Public Member Functions

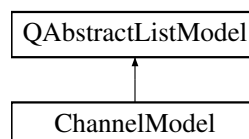
- **ChannelList** (QObject *parent=nullptr)
- Q_INVOKABLE QVector< [CChannel](#) * > **get_channels** ()
- [CChannel](#) * **get_channelAt** (int index)
- void **addChannel** ([CChannel](#) *channel)
- void **removeChannel** ([CChannel](#) *channel)
- void **setItem** ([CChannel](#) *channel)
- bool **setItemAt** (int index, [CChannel](#) *item)

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

- LockVox/Client/includes/channellist.h
- LockVox/Client/channellist.cpp

6.6 ChannelModel Class Reference

Inheritance diagram for ChannelModel:



Public Types

- enum **ModelRoles** { **NameRole** = Qt::UserRole+1 , **nbUsersRole** , **maxUsersRole** }

Public Slots

- void **do_update** ()

Signals

- void **currentIndexChanged** (int index)

Public Member Functions

- **ChannelModel** (QObject *parent=nullptr)
- int **rowCount** (const QModelIndex &parent=QModelIndex()) const override
- QVariant **data** (const QModelIndex &index, int role=Qt::DisplayRole) const override
- bool **setData** (const QModelIndex &index, const QVariant &value, int role=Qt::EditRole) override
- Qt::ItemFlags **flags** (const QModelIndex &index) const override
- bool **insertRows** (int row, int count, const QModelIndex &parent=QModelIndex()) override
- bool **removeRows** (int row, int count, const QModelIndex &parent=QModelIndex()) override
- virtual QHash< int, QByteArray > **roleNames** () const override
- [ChannelList](#) * **getChannelsList** () const
- void **setChannelsList** ([ChannelList](#) *channelsList)

Properties

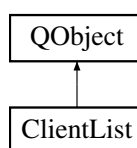
- [ChannelList](#) * **m_channelsList**

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

- LockVox/Client/includes/channelmodel.h
- LockVox/Client/channelmodel.cpp

6.7 ClientList Class Reference

Inheritance diagram for ClientList:



Public Slots

- void **appendItem** ()
- void **removeItem** (int index)

Signals

- void **dataChanged** ()
- void **preItemAppended** ()
- void **postItemAppended** ()
- void **preItemRemoved** (int index)
- void **postItemRemoved** ()

Public Member Functions

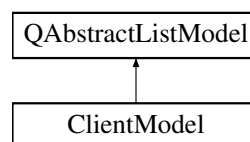
- **ClientList** (QObject *parent=nullptr)
- Q_INVOKABLE QVector< CClient * > **get_clients** ()
- void **addClient** (CClient *client)
- void **removeClient** (CClient *client)
- void **setItem** (CClient *c)
- bool **setItemAt** (int index, CClient *item)

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

- LockVox/Client/includes/clientlist.h
- LockVox/Client/clientlist.cpp

6.8 ClientModel Class Reference

Inheritance diagram for ClientModel:



Public Types

- enum **ModelRoles** {
PseudoRole = Qt::UserRole+1 , **isOnlineRole** , **imageRole** , **descriptionRole** ,
uuidRole }

Public Slots

- void **do_update** ()

Public Member Functions

- **ClientModel** (QObject *parent=nullptr)
- int **rowCount** (const QModelIndex &parent=QModelIndex()) const override
- QVariant **data** (const QModelIndex &index, int role=Qt::DisplayRole) const override
- bool **setData** (const QModelIndex &index, const QVariant &value, int role=Qt::EditRole) override
- Qt::ItemFlags **flags** (const QModelIndex &index) const override
- bool **insertRows** (int row, int count, const QModelIndex &parent=QModelIndex()) override
- bool **removeRows** (int row, int count, const QModelIndex &parent=QModelIndex()) override
- virtual QHash< int, QByteArray > **roleNames** () const override
- CClientList * **getClientList** () const
- void **setClientsList** (CClientList *clientsList)

Properties

- [ClientList](#) * **m_clientsList**

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

- LockVox/Client/includes/clientmodel.h
- LockVox/Client/clientmodel.cpp

6.9 CMessage Class Reference

Declaration of the class [CMessage](#).

Public Member Functions

- **CMessage** (QString from, QString to, QString message, bool isPrivate)
- **CMessage** (QString from, QString to, QString message, QImage img, bool isPrivate)
- **CMessage** (QString xml)
- QString **get_from** ()
- QString **get_to** ()
- QString **get_message** ()
- QString **get_time** ()
- QString **get_date** ()
- QImage **get_image** ()
- bool **get_isPrivate** ()
- QString **get_from_pseudo** ()
- QDomDocument **get_xmlmessage** ()
- void **toXML** ()
- QString **toString** ()
- void **getSenderPseudo** (QVector< [CClient](#) * > clients)

6.9.1 Detailed Description

Declaration of the class [CMessage](#).

[CMessage](#) are simply the messages class so that client can communicate between them...

Author

Valentin Monnot (LockVox Team)

Version

0.1.1

Date

2021

Note

Bug

Warning

Remarks

Copyright

GNU GPL-3.0

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

- LockVox/src/includes/[cmessage.h](#)
- LockVox/src/cmessage.cpp

6.10 CookiesManager Class Reference

Declaration of the class [CookiesManager](#).

Public Member Functions

- **CookiesManager** (QList< [CClient](#) * > clients)
- void **assignCookie** (QUuid id)
- void **deleteCookie** (QUuid id)
- [CSessionCookie](#) **getSessionCookie** (QUuid id)

6.10.1 Detailed Description

Declaration of the class [CookiesManager](#).

This class is used to manage users session cookie

Author

Hugues Mattei (LockVox Team)

Version

0.1.1

Date

2021

Note

Bug

Warning

Remarks

Copyright

GNU GPL-3.0

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

- LockVox/Server/includes/[cookiesmanager.h](#)
- LockVox/Server/cookiesmanager.cpp

6.11 CPacket Class Reference

Declaration of the class Server [CPacket](#).

Public Member Functions

- **CPacket** (QByteArray data, [CClient](#) *client)
- **CPacket** (QString type, QString action)
- QString **GetType** ()
- QString **GetAction** ()
- [CClient](#) **GetSender** ()
- QJsonDocument **GetData** ()
- QByteArray **GetByteArray** ()
- QUuid **get_IdClient** ()
- int **get_IdChannel** ()
- struct [register_info](#) **get_RegisterInfo** ()
- void **SetType** (QString p_type)
- void **SetAction** (QString p_action)
- void **Serialize** ()
- void **Deserialize** ()
- QByteArray **Serialize** ([CServer](#) *c)
- QByteArray **Serialize** (bool isActionValid)

- void **Serialize_newClient** (CClient *client)
- void **Serialize_newChannel** (CChannel *channel)
- void **Serialize_myClient** (CClient *client)
- void **Serialize_auth** (CClient *info, int code)
- void **Serialize_authReq** (QString email, QString pass)
- void **Serialize_ID** (int chan, QUuid client)
- void **Serialize_regReq** (QString username, QString mail, QString password, QString password_confirm)
- void **Serialize_regAns** (int code)
- void **Serialize_MessageListInfo** (int index)
- void **Serialize_Message** (CMessage msg)
- void **Serialize_MessageList** (QList< CMessage > list)
- void **Serialize_messageRequest** (int id, int nb_msg_to_sync, int start_index)
- void **Serialize_messageRequest** (QUuid id, int nb_msg_to_sync, int start_index)
- void **Serialize_ppRequest** (QUuid uuid)
- CClient * **Deserialize_myClient** ()
- CClient * **Deserialize_newClient** ()
- CChannel * **Deserialize_newChannel** ()
- QList< QString > **Deserialize_auth** ()
- void **Deserialize_ID** ()
- void **Deserialize_regReq** ()
- int **Deserialize_regAns** ()
- int **Deserialize_MessageListInfo** ()
- CMessage **Deserialize_Message** ()
- QVector< CMessage > **Deserialize_MessageList** ()
- QList< QString > **deserialize_messageRequest** ()
- int **Deserialize_MessageError** ()
- QList< QString > **Deserialize_ppAns** ()
- void **Serialize_Register** (struct register_info reg)
- struct register_info **Deserialize_Register** ()
- CClient * **Deserialize_authAns** ()
- CPacket (QByteArray data, CClient *client)
- CPacket (QString type, QString action)
- QString **GetType** ()
- QString **GetAction** ()
- QJsonDocument **GetData** ()
- QByteArray **GetByteArray** ()
- QUuid **get_IdClient** ()
- int **get_IdChannel** ()
- void **SetType** (QString p_type)
- void **SetAction** (QString p_action)
- void **Serialize** ()
- void **Serialize** (CServer *c)
- void **Serialize_newClient** (CClient *client, bool sendPP)
- void **Serialize_newChannel** (CChannel *channel)
- void **Serialize_myClient** (CClient *client)
- void **Serialize_auth** (CClient *info, int code)
- void **Serialize_ID** (int chan, QUuid client)
- void **Serialize_regReq** (QString username, QString mail, QString password, QString password_confirm)
- void **Serialize_regAns** (int code)
- void **Serialize_MessageListInfo** (int index)
- void **Serialize_Message** (CMessage msg)
- void **Serialize_MessageList** (QList< CMessage > list)
- void **Serialize_MessageError** (int code)
- void **Serialize_ppAnswer** (QImage pp, QUuid uuid)
- CClient * **Deserialize_myClient** ()

- [CClient](#) * **Deserialize_newClient** ()
- [CChannel](#) * **Deserialize_newChannel** ()
- QList< QString > **Deserialize_auth** ()
- void **Deserialize_ID** ()
- QList< QString > **Deserialize_regReq** ()
- [CMessage](#) **Deserialize_Message** ()
- QList< [CMessage](#) > **Deserialize_MessageList** ()
- QList< QString > **deserialize_messageRequest** ()
- QString **deserialize_ppRequest** ()
- void **Serialize_Register** (struct [register_info](#) reg)
- struct [register_info](#) **Deserialize_Register** ()

6.11.1 Detailed Description

Declaration of the class Server [CPacket](#).

[CPacket](#) are used to discuss between client and server

Author

LockVox Team

Version

0.1.1

Date

2021

Note

Bug

Warning

Remarks

Copyright

GNU GPL-3.0

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

- LockVox/Client/includes/cpacket.h
- LockVox/Client/cpacket.cpp

6.12 crole Class Reference

Public Member Functions

- **crole** ([crole](#) *p_father)
- void **AddFather** ([crole](#) *p_father)
- void **AddChild** ([crole](#) *p_child)
- void **DelChild** (QString p_name)
- void **DelChild** (int p_id)
- void **SetWrite** (bool v)
- void **SetSpeak** (bool v)
- void **SetKick** (bool v)
- void **SetDelete** (bool v)
- void **SetSee** (bool v)
- void **SetMute** (bool v)
- bool **GetWrite** ()
- bool **GetSpeak** ()
- bool **GetKick** ()
- bool **GetDelete** ()
- bool **GetSee** ()
- bool **GetMute** ()

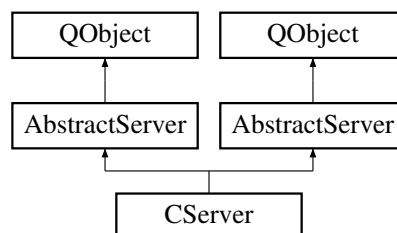
The documentation for this class was generated from the following files:

- LockVox/src/includes/crole.h
- LockVox/src/crole.cpp

6.13 CServer Class Reference

Declaration of the class [CServer](#).

Inheritance diagram for CServer:



Public Slots

- void **RequestServer** (int type, int action, [CClient](#) *client, [CChannel](#) *chan)
- bool **Login** (QString mail, QString passwd)
- bool **Register** (QString username, QString mail, QString password, QString password_confirm)
- bool **sendMessage** (QString msg)
- bool **sendMessage** (QString msg, QUuid id)
- void **connectServer** (QString ip)
- void **connectServer** ()
- void **disconnectServer** ()
- bool **changeUserName** (QString pseudo)
- bool **changeEmail** (QString email)
- bool **changeDescription** (QString description)

Signals

- void **changeState** (QString)
- void **selfChanged** (CClient *)
- void **connected** ()
- void **disconnected** ()
- void **picturesLoad** ()
- void **sendToAll** (QByteArray out)

Public Member Functions

- CServer ()
The CServer constructor.
- CServer (ClientList *clients, ChannelList *channels)
- QTcpSocket * **get_socket** ()
- void **set_socket** (QTcpSocket *soc)
- ClientList * **getClientsList** () const
- void **setClientsList** (ClientList *clientsList)
- ChannelList * **getChannelsList** ()
- void **setChannelsList** (ChannelList *channelsList)
- CClient * **get_self** ()
- void **set_self** (CClient *c)
- int **getCurrentChannelIndex** () const
- void **setCurrentChannelIndex** (int currentChannelIndex)
- MessageList * **getMessagesList** () const
- void **setMessagesList** (MessageList *messagesList)
- QString **getName** () const
- void **setName** (const QString &name)
- QString **getIp** () const
- void **setIp** (const QString &ip)
- void **processIncomingData** (QByteArray data)
- void **loadAllCompenent** ()
- void **loadChannelsClientsList** ()
- void **loadMessages** ()
- void **checkCompenents** ()
- void **checkFinishLoad** ()
- void **changeChannel** (int id)
- void **quitChannel** (int id)
- void **joinChannel** (int id)
- QByteArray **Serialize** ()
- void **Deserialize** (QByteArray in)
- QByteArray **SerializeChannels** ()
- QByteArray **SerializeClients** ()
- void **DeserializeChannels** (QByteArray in)
- void **DeserializeClient** (QByteArray in)
- void **deserializeChannel** (QJsonArray &json_array)
- void **deserializeClients** (QJsonArray &json_array)
- CChannel * **deserializeToChannel** (QJsonObject json_obj)
- CClient * **deserializeToClient** (QJsonObject json_obj)
- void **start** ()

This function should be called to start the server. Starting TCP Server and initialising DB connection.

- CDatabase * **get_database** ()
- void **set_database** (CDatabase *db)

- [CClient](#) * **whichClient** (QTcpSocket *soc)
- void **sendToClient** (QByteArray out, [CClient](#) *client)
- void **sendToAllExceptClient** (QByteArray out, [CClient](#) *client)
- void **AddChannel** ([CChannel](#) *channel)
- void **AddBannedUser** ([CClient](#) *client)
- void **RemoveBannedUser** ([CClient](#) *client)
- QList< [CClient](#) * > **GetBannedUserList** ()
- void **writeToLog** (QString error, int level)
- void **connectClient** ([CClient](#) *client)

Additional Inherited Members

6.13.1 Detailed Description

Declaration of the class [CServer](#).

Here is the main process of a LockVox server

Author

LockVox Team

Version

0.2.0

Date

2021

Note

Bug

Warning

Remarks

Copyright

GNU GPL-3.0

6.13.2 Constructor & Destructor Documentation

6.13.2.1 CServer()

```
CServer::CServer ( )
```

The [CServer](#) constructor.

See also

[CServer::CServer](#)

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

- LockVox/Client/includes/cserver.h
- LockVox/Client/cserver.cpp

6.14 CSessionCookie Class Reference

Declaration of the class [CSessionCookie](#).

Public Member Functions

- **CSessionCookie** ([CClient](#) *c)
- QString **getCookie** () const
- void **setCookie** (const QString &cookie)
- QString **HighRandomStringGenerator** ()
- void **clearCookie** ()

6.14.1 Detailed Description

Declaration of the class [CSessionCookie](#).

This class represent a cookie session

Author

Hugues Mattei (LockVox Team)

Version

0.1.1

Date

2021

Note

Bug

Warning

Une mauvaise utilisation peut faire planter votre application (c'est votre faute)

Remarks

Les destructeur non définie ça emepche de compiler !

Copyright

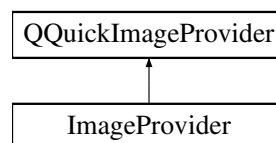
GNU GPL-3.0

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

- LockVox/Server/includes/[csessioncookie.h](#)
- LockVox/src/csessioncookie.cpp

6.15 ImageProvider Class Reference

Inheritance diagram for ImageProvider:



Public Member Functions

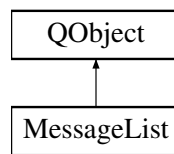
- **ImageProvider** ([ClientList](#) *clientList, QImage img)
- QImage **requestImage** (const QString &id, QSize *size, const QSize &requestedSize)
- [ClientList](#) * **clientsList** () const
- void **setClientsList** ([ClientList](#) *clientsList)

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

- LockVox/imageprovider.h
- LockVox/imageprovider.cpp

6.16 MessageList Class Reference

Inheritance diagram for MessageList:



Public Slots

- void **appendItem** ()
- void **removeItem** (int index)

Signals

- void **dataChanged** ()
- void **preItemAppended** ()
- void **postItemAppended** ()
- void **preItemRemoved** (int index)
- void **postItemRemoved** ()
- void **beginChangeList** ()
- void **endChangeList** ()
- void **listChanged** ([MessageList](#) *msgList)

Public Member Functions

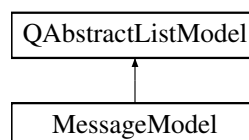
- **MessageList** (QObject *parent=nullptr)
- Q_INVOKABLE QVector< [CMessage](#) > **get_messages** ()
- void **set_messages** (QVector< [CMessage](#) > messages)
- void **addMessage** ([CMessage](#) client)
- void **removeMessage** ([CMessage](#) client)
- void **setItem** ([CMessage](#) c)
- bool **setItemAt** (int index, [CMessage](#) item)
- bool **getHasBeenLoad** () const
- void **setHasBeenLoad** (bool hasBeenLoad)

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

- LockVox/Client/includes/messagelist.h
- LockVox/Client/messagelist.cpp

6.17 MessageModel Class Reference

Inheritance diagram for MessageModel:



Public Types

- enum **ModelRoles** {
FromRole = Qt::UserRole+1 , **ToRole** , **TimeRole** , **DateRole** ,
MessageRole }

Public Slots

- void **do_update** ()
- void **onListChanged** ([MessageList](#) *msgList)

Public Member Functions

- MessageModel** (QObject *parent=nullptr)
- int **rowCount** (const QModelIndex &parent=QModelIndex()) const override
- QVariant **data** (const QModelIndex &index, int role=Qt::DisplayRole) const override
- bool **setData** (const QModelIndex &index, const QVariant &value, int role=Qt::EditRole) override
- Qt::ItemFlags **flags** (const QModelIndex &index) const override
- bool **insertRows** (int row, int count, const QModelIndex &parent=QModelIndex()) override
- bool **removeRows** (int row, int count, const QModelIndex &parent=QModelIndex()) override
- virtual QHash< int, QByteArray > **roleNames** () const override
- [MessageList](#) * **getMessagesList** () const
- void **setMessagesList** ([MessageList](#) *messagesList)

Properties

- [MessageList](#) * **m_messagesList**

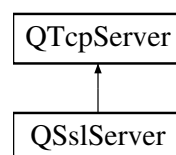
The documentation for this class was generated from the following files:

- LockVox/Client/includes/messagemodel.h
- LockVox/Client/messagemodel.cpp

6.18 QSslServer Class Reference

An extension of QTcpServer to support ssl. The counterpart to QSslSocket.

Inheritance diagram for QSslServer:



Public Member Functions

- **QSslServer** (QObject *parent=nullptr)
- QSslSocket * **nextPendingConnection** () override
- QSslConfiguration **sslConfiguration** () const
- void **setSslConfiguration** (QSslConfiguration configuration)
- void **setLocalCertificateChain** (const QList< QSslCertificate > &localChain)
- void **setLocalCertificateChain** (const QString &fileName, QSsl::EncodingFormat format=QSsl::Pem, QSslCertificate::PatternSyntax syntax=QSslCertificate::PatternSyntax::FixedString)
- QList< QSslCertificate > **localCertificateChain** () const
- void **setLocalCertificate** (const QSslCertificate &certificate)
- void **setLocalCertificate** (const QString &fileName, QSsl::EncodingFormat format=QSsl::Pem)
- QSslCertificate **localCertificate** () const
- void **setPrivateKey** (const QSslKey &key)
- void **setPrivateKey** (const QString &fileName, QSsl::KeyAlgorithm algorithm=QSsl::Rsa, QSsl::EncodingFormat format=QSsl::Pem, const QByteArray &passPhrase=QByteArray{})
- QSslKey **privateKey** () const
- bool **addCaCertificates** (const QString &path, QSsl::EncodingFormat format=QSsl::Pem, QSslCertificate::PatternSyntax syntax=QSslCertificate::PatternSyntax::FixedString)
- void **addCaCertificate** (const QSslCertificate &certificate)
- void **addCaCertificates** (const QList< QSslCertificate > &certificates)
- bool **importPkcs12** (const QString &path, const QByteArray &passPhrase=QByteArray(), bool replaceCertificates=false)
- QSsl::SslProtocol **protocol** () const
- void **setProtocol** (QSsl::SslProtocol protocol)

Protected Member Functions

- void **incomingConnection** (qintptr handle) override

6.18.1 Detailed Description

An extension of QTcpServer to support ssl. The counterpart to QSslSocket.

Author

Skycoder42 1.1.0 updated by Valentin Monnot 2021 BSD-3-Clause License

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

- LockVox/Server/includes/[qsslserver.h](#)
- LockVox/Server/qsslserver.cpp

6.19 register_info Struct Reference

Public Attributes

- QString **email**
- QString **password**
- QString **password_confirm**
- QString **name**

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

- LockVox/Client/includes/cpacket.h

6.20 SHA256 Class Reference

sha256 C implementation

Public Member Functions

- void **init** ()
- void **update** (const unsigned char *message, unsigned int len)
- void **final** (unsigned char *digest)

Static Public Attributes

- static const unsigned int **DIGEST_SIZE** = (256 / 8)

Protected Types

- typedef unsigned char **uint8**
- typedef unsigned int **uint32**
- typedef unsigned long long **uint64**

Protected Member Functions

- void **transform** (const unsigned char *message, unsigned int block_nb)

Protected Attributes

- unsigned int **m_tot_len**
- unsigned int **m_len**
- unsigned char **m_block** [2 *SHA224_256_BLOCK_SIZE]
- uint32 **m_h** [8]

Static Protected Attributes

- static const uint32 **sha256_k** []
- static const unsigned int **SHA224_256_BLOCK_SIZE** = (512/8)

6.20.1 Detailed Description

sha256 C implementation

Author

Oliver Gay updated to C++ by Zedwood 2012 BSD

6.20.2 Member Data Documentation

6.20.2.1 sha256_k

```
const unsigned int SHA256::sha256_k [static], [protected]
```

Initial value:

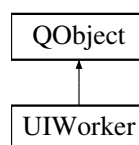
```
=
{ 0x428a2f98, 0x71374491, 0xb5c0fbcf, 0xe9b5dba5,
  0x3956c25b, 0x59f111f1, 0x923f82a4, 0xab1c5ed5,
  0xd807aa98, 0x12835b01, 0x243185be, 0x550c7dc3,
  0x72be5d74, 0x80deb1fe, 0x9bdc06a7, 0xc19bf174,
  0xe49b69c1, 0xefbe4786, 0x0fc19dc6, 0x240ca1cc,
  0x2de92c6f, 0x4a7484aa, 0x5cb0a9dc, 0x76f988da,
  0x983e5152, 0xa831c66d, 0xb00327c8, 0xbf597fc7,
  0xc6e00bf3, 0xd5a79147, 0x06ca6351, 0x14292967,
  0x27b70a85, 0x2e1b2138, 0x4d2c6dfe, 0x53380d13,
  0x650a7354, 0x766a0abb, 0x81c2c92e, 0x92722c85,
  0xa2bfe8a1, 0xa81a664b, 0xc24b8b70, 0xc76c51a3,
  0xd192e819, 0xd6990624, 0xf40e3585, 0x106aa070,
  0x19a4c116, 0x1e376c08, 0x2748774c, 0x34b0bcb5,
  0x391c0cb3, 0x4ed8aa4a, 0x5b9cca4f, 0x682e6fff3,
  0x748f82ee, 0x78a5636f, 0x84c87814, 0x8cc70208,
  0x90befffa, 0xa4506ceb, 0xbef9a3f7, 0xc67178f2}
```

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

- LockVox/Server/includes/[sha256.h](#)
- LockVox/Server/sha256.cpp

6.21 UIWorker Class Reference

Inheritance diagram for UIWorker:



Public Slots

- void **onChangeState** (QString)
- void **onSelfChanged** (CClient *)
- void **onCurrentIndexChanged** (int index)
- void **onPicturesLoad** ()
- void **onConnected** ()
- void **onDisconnected** ()
- void **onQuit** ()
- void **onDisconnect** ()
- void **onChangeServer** ()
- void **onConfirmQuit** (int)

Signals

- void **changeState** (QString newState)
- void **connect_server** (QString)
- void **login_request** (QString, QString)
- void **change_username** (QString)

Public Member Functions

- **UIWorker** (QGuiApplication *app)

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

- LockVox/Client/includes/uiworker.h
- LockVox/Client/uiworker.cpp

Chapter 7

File Documentation

7.1 LockVox/Client/cclient.cpp File Reference

Definition of class [CClient](#).

```
#include "cclient.h"
```

7.1.1 Detailed Description

Definition of class [CClient](#).

Author

Version

Date

7.2 LockVox/Server/cclient.cpp File Reference

Definition of class [CClient](#).

```
#include "cclient.h"
```

7.2.1 Detailed Description

Definition of class [CClient](#).

Author

LockVox Team

Version

0.2.0

Date

2021

7.3 LockVox/Client/includes/cclient.h File Reference

Declaration of the class [CClient](#).

```
#include <QString>
#include <QList>
#include <QDebug>
#include <QObject>
#include <QtNetwork>
#include <QUuid>
#include <QImage>
#include "cchannel.h"
```

Classes

- class [CClient](#)

Declaration of the class [CClient](#).

7.3.1 Detailed Description

Declaration of the class [CClient](#).

This class represent an user

Author

LockVox Team

Version

0.1.1

Date

2021

Copyright

GNU GPL-3.0

7.4 LockVox/Server/includes/cclient.h File Reference

Declaration of the class [CClient](#).

```
#include <QString>
#include <QList>
#include <QDebug>
#include <QObject>
#include <QtNetwork>
#include <QUuid>
#include <QImage>
#include "cpacket.h"
#include "sha256.h"
```

Classes

- class [CClient](#)

Declaration of the class [CClient](#).

Macros

- #define **USER** 0
- #define **SERVER** 1
- #define **SERVER_WARN** 2
- #define **SERVER_ERR** 3
- #define **DB_ERR** 4

7.4.1 Detailed Description

Declaration of the class [CClient](#).

This class represent an user

Author

LockVox Team

Version

0.2.0

Date

2021

Copyright

GNU GPL-3.0

7.5 LockVox/Server/includes/cpacket.h File Reference

Declaration of the class Server [CPacket](#).

```
#include "cmessage.h"
#include "cclient.h"
#include "cchannel.h"
#include <QString>
#include <QList>
```

Classes

- class [CPacket](#)

Declaration of the class Server [CPacket](#).

7.5.1 Detailed Description

Declaration of the class Server [CPacket](#).

This class used for LockVox protocol

Author

LockVox Team

Version

0.1.1

Date

2021

Copyright

GNU GPL-3.0

7.6 LockVox/Server/includes/cserver.h File Reference

Declaration of the class [CServer](#).

```
#include <QDebug>
#include <QRegExp>
#include <QDir>
#include <QRandomGenerator>
#include "AbstractServer.h"
#include "cdatabase.h"
#include "cpacket.h"
#include "caudio.h"
#include "Server/config.h"
```

Classes

- class [CServer](#)

Declaration of the class [CServer](#).

7.6.1 Detailed Description

Declaration of the class [CServer](#).

This class the main class of LockVox server

Author

LockVox Team

Version

0.2.0

Date

2021

Copyright

GNU GPL-3.0

7.7 LockVox/Server/includes/cdatabase.h File Reference

Declaration of the class [CDatabase](#).

```
#include <iostream>
#include <iomanip>
#include <sstream>
#include <string>
#include <QApplication>
#include "mysql.h"
#include "sha256.h"
#include "cchannel.h"
#include "cclient.h"
```

Classes

- class [CDatabase](#)

SQL Database Class.

7.7.1 Detailed Description

Declaration of the class [CDatabase](#).

This class represent an user

Author

Valentin Monnot (LockVox Team)

Version

0.2.0

Date

2021

Copyright

GNU GPL-3.0

7.8 LockVox/Server/includes/cookiesmanager.h File Reference

Declaration of the class CookieManager.

```
#include <QMap>
#include "cclient.h"
#include "csessioncookie.h"
```

Classes

- class [CookiesManager](#)

Declaration of the class [CookiesManager](#).

7.8.1 Detailed Description

Declaration of the class CookieManager.

This class is used to manage users session cookie

Author

LockVox Team

Version

0.1.1

Date

2021

Copyright

GNU GPL-3.0

7.9 LockVox/Server/includes/csessioncookie.h File Reference

Declaration of the class [CSessionCookie](#).

```
#include "cclient.h"
```

Classes

- class [CSessionCookie](#)

Declaration of the class [CSessionCookie](#).

7.9.1 Detailed Description

Declaration of the class [CSessionCookie](#).

This class represent a cookie session

Author

Hugues Mattei (LockVox Team)

Version

0.1.1

Date

2021

Copyright

GNU GPL-3.0

7.10 LockVox/Server/includes/qsslserver.h File Reference

An extension of QTcpServer to support ssl. The counterpart to QSslSocket.

```
#include <QTcpServer>
#include <QSslSocket>
#include <QSslConfiguration>
```

Classes

- class [QSslServer](#)

An extension of QTcpServer to support ssl. The counterpart to QSslSocket.

7.10.1 Detailed Description

An extension of QTcpServer to support ssl. The counterpart to QSslSocket.

Author

Skycoder42 1.1.0 updated by Valentin Monnot 2021 BSD-3-Clause License

7.11 LockVox/Server/includes/sha256.h File Reference

sha256 C implementation

```
#include <string>
```

Classes

- class [SHA256](#)
sha256 C implementation

Macros

- #define **SHA2_SHFR**(x, n) (x >> n)
- #define **SHA2_ROTTR**(x, n) ((x >> n) | (x << ((sizeof(x) << 3) - n)))
- #define **SHA2_ROTTL**(x, n) ((x << n) | (x >> ((sizeof(x) << 3) - n)))
- #define **SHA2_CH**(x, y, z) ((x & y) ^ (~x & z))
- #define **SHA2_MAJ**(x, y, z) ((x & y) ^ (x & z) ^ (y & z))
- #define **SHA256_F1**(x) (SHA2_ROTTR(x, 2) ^ SHA2_ROTTR(x, 13) ^ SHA2_ROTTR(x, 22))
- #define **SHA256_F2**(x) (SHA2_ROTTR(x, 6) ^ SHA2_ROTTR(x, 11) ^ SHA2_ROTTR(x, 25))
- #define **SHA256_F3**(x) (SHA2_ROTTR(x, 7) ^ SHA2_ROTTR(x, 18) ^ SHA2_SHFR(x, 3))
- #define **SHA256_F4**(x) (SHA2_ROTTR(x, 17) ^ SHA2_ROTTR(x, 19) ^ SHA2_SHFR(x, 10))
- #define **SHA2_UNPACK32**(x, str)
- #define **SHA2_PACK32**(str, x)

Functions

- std::string **sha256** (std::string input)

7.11.1 Detailed Description

sha256 C implementation

Author

Oliver Gay updated to C++ by Zedwood 2012 BSD

7.11.2 Macro Definition Documentation

7.11.2.1 SHA2_PACK32

```
#define SHA2_PACK32(  
    str,  
    x )
```

Value:

```
{  
    * (x) =    ((uint32) *((str) + 3)      )  \\  
              | ((uint32) *((str) + 2) << 8)  \\  
              | ((uint32) *((str) + 1) << 16)  \\  
              | ((uint32) *((str) + 0) << 24);  \\  
}
```

7.11.2.2 SHA2_UNPACK32

```
#define SHA2_UNPACK32(  
    x,  
    str )
```

Value:

```
{  
    *((str) + 3) = (uint8) ((x)      );  \\  
    *((str) + 2) = (uint8) ((x) >> 8);  \\  
    *((str) + 1) = (uint8) ((x) >> 16);  \\  
    *((str) + 0) = (uint8) ((x) >> 24);  \\  
}
```

7.12 LockVox/src/includes/AbstractServer.h File Reference

Declaration of the class [AbstractServer](#).

```
#include <QDebug>  
#include <QObject>  
#include <QtNetwork>  
#include <QUuid>  
#include "cchannel.h"  
#include "cclient.h"
```

Classes

- class [AbstractServer](#)

Declaration of the class [AbstractServer](#).

7.12.1 Detailed Description

Declaration of the class [AbstractServer](#).

This class represent the virtual class of [CServer](#)

Author

LockVox Team

Version

0.2.0

Date

2021

Copyright

GNU GPL-3.0

7.13 LockVox/src/includes/cchannel.h File Reference

Declaration of the class [CChannel](#).

```
#include <QJsonObject>
#include <QUuid>
#include "cclient.h"
#include "Client/includes/messagelist.h"
```

Classes

- class [CChannel](#)

Declaration of the class [CChannel](#).

7.13.1 Detailed Description

Declaration of the class [CChannel](#).

This class represent a channel

Author

LockVox Team

Version

0.1.1

Date

2021

Copyright

GNU GPL-3.0

7.14 LockVox/src/includes/cmessage.h File Reference

Declaration of the class [CMessage](#).

```
#include <QString>
#include <QtXml>
#include <QTime>
#include <QDate>
#include <QFile>
#include <QImage>
#include <QDebug>
#include "cclient.h"
#include "channel.h"
```

Classes

- class [CMessage](#)

Declaration of the class [CMessage](#).

7.14.1 Detailed Description

Declaration of the class [CMessage](#).

[CMessage](#) are simply the messages class so that client can communicate between them...

Author

Valentin Monnot (LockVox Team)

Version

0.1.1

Date

2021

Copyright

GNU GPL-3.0

Index

- ~CClient
 - CClient, [27](#)
- AbstractServer, [11](#)
- addUser
 - CChannel, [16](#)
- CChannel, [13](#)
 - addUser, [16](#)
 - CChannel, [14–16](#)
 - delUser, [16](#)
 - deserialize, [17](#)
 - get_clients, [17](#)
 - get_id, [17](#)
 - get_maxUsers, [18](#)
 - get_name, [18](#)
 - get_nbClients, [18](#)
 - getMessagesLists, [19](#)
 - serializeToObj, [19](#)
 - set_all, [19](#)
 - set_clients, [19](#)
 - set_id, [20](#)
 - set_maxUsers, [20](#)
 - set_name, [21](#)
 - set_nbClients, [21](#)
 - setMessagesLists, [21](#)
- CClient, [22](#)
 - ~CClient, [27](#)
 - CClient, [25–27](#)
 - createChannelIndex, [27](#)
 - createMessageList, [28](#)
 - deserialize, [28](#)
 - get_description, [29](#)
 - get_idChannel, [29](#)
 - get_isAuthenticate, [29](#)
 - get_isOnline, [30](#)
 - get_mail, [30](#)
 - get_profilePic, [30](#)
 - get_pseudo, [31](#)
 - get_socket, [31](#)
 - get_uuid, [31](#)
 - insertChannelIndex, [32](#)
 - processData, [32](#)
 - readChannelIndex, [32](#)
 - serializeToObj, [33](#)
 - set_all, [33](#)
 - set_description, [33](#)
 - set_idChannel, [34](#)
 - set_isAuthenticate, [34](#)
 - set_isOnline, [34](#)
 - set_mail, [35](#)
 - set_profilePic, [35](#)
 - set_pseudo, [36](#)
 - set_socket, [36](#)
 - set_uuid, [36](#)
- CDatabase, [37](#)
- ChannelList, [38](#)
- ChannelModel, [39](#)
- ClientList, [40](#)
- ClientModel, [41](#)
- CMessage, [42](#)
- CookiesManager, [43](#)
- CPacket, [44](#)
- createChannelIndex
 - CClient, [27](#)
- createMessageList
 - CClient, [28](#)
- crole, [47](#)
- CServer, [47](#)
 - CServer, [50](#)
- CSessionCookie, [50](#)
- delUser
 - CChannel, [16](#)
- deserialize
 - CChannel, [17](#)
 - CClient, [28](#)
- get_clients
 - CChannel, [17](#)
- get_description
 - CClient, [29](#)
- get_id
 - CChannel, [17](#)
- get_idChannel
 - CClient, [29](#)
- get_isAuthenticate
 - CClient, [29](#)
- get_isOnline
 - CClient, [30](#)
- get_mail
 - CClient, [30](#)
- get_maxUsers
 - CChannel, [18](#)
- get_name
 - CChannel, [18](#)
- get_nbClients
 - CChannel, [18](#)
- get_profilePic
 - CClient, [30](#)

get_pseudo
 CClient, 31
get_socket
 CClient, 31
get_uuid
 CClient, 31
getMessagesLists
 CChannel, 19

ImageProvider, 51
insertChannelIndex
 CClient, 32

LockVox/Client/cclient.cpp, 59
LockVox/Client/includes/cclient.h, 60
LockVox/Server/cclient.cpp, 59
LockVox/Server/includes/cclient.h, 61
LockVox/Server/includes/cdatabase.h, 63
LockVox/Server/includes/cookiesmanager.h, 64
LockVox/Server/includes/cpacket.h, 62
LockVox/Server/includes/cserver.h, 62
LockVox/Server/includes/csessioncookie.h, 65
LockVox/Server/includes/qsslserver.h, 65
LockVox/Server/includes/sha256.h, 66
LockVox/src/includes/AbstractServer.h, 67
LockVox/src/includes/cchannel.h, 68
LockVox/src/includes/cmessage.h, 69

MessageList, 52
MessageModel, 52

processData
 CClient, 32

QSSLServer, 53

readChannelIndex
 CClient, 32
register_info, 54

serializeToObj
 CChannel, 19
 CClient, 33
set_all
 CChannel, 19
 CClient, 33
set_clients
 CChannel, 19
set_description
 CClient, 33
set_id
 CChannel, 20
set_idChannel
 CClient, 34
set_isAuthenticate
 CClient, 34
set_isOnline
 CClient, 34
set_mail
 CClient, 35

set_maxUsers
 CChannel, 20
set_name
 CChannel, 21
set_nbClients
 CChannel, 21
set_profilePic
 CClient, 35
set_pseudo
 CClient, 36
set_socket
 CClient, 36
set_uuid
 CClient, 36
setMessagesLists
 CChannel, 21
SHA256, 55
 sha256_k, 56
sha256.h
 SHA2_PACK32, 67
 SHA2_UNPACK32, 67
sha256_k
 SHA256, 56
SHA2_PACK32
 sha256.h, 67
SHA2_UNPACK32
 sha256.h, 67

UIWorker, 56