

MessageAPI

2.0

Generated by Doxygen 1.7.5.1

Sun Oct 30 2011 11:53:50

Contents

Chapter 1

Class Index

1.1 Class List

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

BroadcastService	??
UdpMessage	??
UdpUser	??

Chapter 2

Class Documentation

2.1 BroadcastService Class Reference

Signals

- void [OnServiceStarted](#) (bool)
BroadcastService Started.
- void [OnMessageRecieved](#) (UdpMessage *)
Message Recieved.
- void [OnMessageFail](#) ()
Message Send Failed.
- void [OnMessageSuccess](#) ()
Message Send Succeeded.

Public Member Functions

- [BroadcastService](#) (QString, qint16)
Broadcast Service Constructor.
- bool [startService](#) ()
Starts the Broadcast Service.
- void [sendMessage](#) (QString, [UdpMessage::UdpMessageType](#))
Send a message.
- void [sendMessage](#) ([UdpMessage](#), QString)
Send a message.
- void [sendMessage](#) ([UdpMessage](#) *)
Send a message.
- void [joinGroup](#) (QString GroupName)
Join a message group.
- void [leaveGroup](#) (QString GroupName)
Leave a message group.

2.1.1 Constructor & Destructor Documentation

2.1.1.1 `BroadcastService::BroadcastService (QString address = "255.255.255.255", quint16 port = 5683)`

Broadcast Service Constructor.

This must be called before attempting to use a new [BroadcastService](#)

Parameters

<i>address</i>	the IP address that the BroadcastService will run on. (default: 255.-255.255.255)
<i>port</i>	the port that the BroadcastService will run on. (default: 5683)

2.1.2 Member Function Documentation

2.1.2.1 `void BroadcastService::joinGroup (QString GroupName)`

Join a message group.

Used to join a message group

Parameters

<i>GroupName</i>	the name of the group to join
------------------	-------------------------------

2.1.2.2 `void BroadcastService::leaveGroup (QString GroupName)`

Leave a message group.

Used to leave a message group

Parameters

<i>GroupName</i>	the name of the group to leave
------------------	--------------------------------

2.1.2.3 `void BroadcastService::OnMessageFail () [signal]`

Message Send Failed.

Emitted when a message send operation fails

2.1.2.4 `void BroadcastService::OnMessageRecieved (UdpMessage *) [signal]`

Message Recieved.

Emitted when the [BroadcastService](#) has received a message

2.1.2.5 void BroadcastService::OnMessageSuccess () [signal]

Message Send Succeeded.

Emitted when a message send operation succeeds

2.1.2.6 void BroadcastService::OnServiceStarted (bool) [signal]

[BroadcastService](#) Started.

Emitted when the [BroadcastService](#) has successfully started

**2.1.2.7 void BroadcastService::sendMessage (QString *message*,
UdpMessage::UdpMessageType *type*)**

Send a message.

Use this function to send a message to everyone.

Parameters

<i>message</i>	the string to be sent
<i>type</i>	a UdpMessage::UdpMessageType specifying what type the message is

**2.1.2.8 void BroadcastService::sendMessage (UdpMessage *message*, QString
alias)**

Send a message.

Use this function to send a message with an alias.

Parameters

<i>message</i>	the UdpMessage to be send
<i>alias</i>	the current user's alias

2.1.2.9 void BroadcastService::sendMessage (UdpMessage * *message*)

Send a message.

Use this function to send a [UdpMessage](#)

Parameters

<i>message</i>	the UdpMessage to send
----------------	--

2.1.2.10 bool BroadcastService::startService ()

Starts the Broadcast Service.

This function is called to start the [BroadcastService](#). It is automatically called by - [BroadcastService::BroadcastService](#)

Returns

A boolean value indicating if the start succeeded.

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

- MessageAPI/broadcastservice.h
- MessageAPI/broadcastservice.cpp

2.2 UdpMessage Class Reference

Public Types

- enum [UdpMessageType](#) { [Unknown](#), [WebsiteShare](#), [Chat](#), [System](#), [Error](#), [Whois](#), [Custom](#) }

Message Type.

Public Member Functions

- [UdpMessage](#) (QString, QString)
UdpMessage Constructor.
- [UdpMessage](#) (QString, QString, [UdpMessageType](#))
UdpMessage Constructor.
- [UdpMessage](#) ([UdpUser](#) &, QString, [UdpMessageType](#))
UdpMessage Constructor.
- [UdpMessageType](#) [messageTypeFromString](#) (QString)
Find UdpMessageType from QString.
- [UdpMessage](#) * [parseMessage](#) (QString)
Parse Message from QString.

Public Attributes

- [UdpUser](#) * [IntendedUser](#)
Intended User.
- [UdpUser](#) * [SenderUser](#)
Sender User.
- QString [Message](#)

Message Data.

- [UdpMessageType](#) [MessageType](#)

Message Type.

- QString [GroupName](#)

Group Name.

- int [ApiLevel](#)

Api Level.

- QList< QString > [CustomFields](#)

Custom Fields.

Static Public Attributes

- static const QChar [controlChar](#) = QChar(65533)

Control Char.

2.2.1 Member Enumeration Documentation

2.2.1.1 enum UdpMessage::UdpMessageType

Message Type.

This enum is used to specify what the type of the [UdpMessage](#) is.

Enumerator:

Unknown Message Type Unknown

WebsiteShare Message Type WebsiteShare

Chat Message Type Chat

System Message Type System

Error Message Type Error

Whois Message Type Whois

Custom Message Type Custom

2.2.2 Constructor & Destructor Documentation

2.2.2.1 UdpMessage::UdpMessage (QString *user*, QString *message*)

[UdpMessage](#) Constructor.

Creates a new [UdpMessage](#).

Parameters

<i>user</i>	the username of the recipient user
<i>message</i>	the message to send

2.2.2.2 `UdpMessage::UdpMessage (QString user, QString message, UdpMessageType type)`

[UdpMessage](#) Constructor.

Creates a new [UdpMessage](#).

Parameters

<i>user</i>	the username of the recipient user
<i>message</i>	the message to send
<i>type</i>	the type of the message

2.2.2.3 `UdpMessage::UdpMessage (UdpUser & user, QString message, UdpMessageType type)`

[UdpMessage](#) Constructor.

Creates a new [UdpMessage](#).

Parameters

<i>user</i>	a UdpUser specifying the recipient of the message
<i>message</i>	the message to send
<i>type</i>	the type of the message

2.2.3 Member Function Documentation

2.2.3.1 `UdpMessage::UdpMessageType UdpMessage::messageTypeFromString (QString type)`

Find UdpMessageType from QString.

This is used to find a [UdpMessage::UdpMessageType](#) from a given string

Parameters

<i>type</i>	the string from which the UdpMessage::UdpMessageType is found
-------------	---

Returns

a [UdpMessage::UdpMessageType](#) from the string. Returns type "Unknown" if the string doesn't match any known types

2.2.3.2 `UdpMessage * UdpMessage::parseMessage (QString message)`

Parse Message from QString.

Takes a Message in string form (such as from a network stream) and parses it into a [UdpMessage](#)

Parameters

<i>message</i>	the string to parse
----------------	---------------------

Returns

the [UdpMessage](#) that was parsed from the string

2.2.4 Member Data Documentation

2.2.4.1 int UdpMessage::ApiLevel

Api Level.

The API Level of the client which created the message

2.2.4.2 const QChar UdpMessage::controlChar = QChar(65533) [static]

Control Char.

The control char is used to separate the individual elements when the [UdpMessage](#) is in string form

2.2.4.3 QList<QString> UdpMessage::CustomFields

Custom Fields.

Fields which can be attached to the message which are not part of the MessageAPI Protocol

2.2.4.4 QString UdpMessage::GroupName

Group Name.

The group to which the message is addressed to.

2.2.4.5 UdpUser* UdpMessage::IntendedUser

Intended User.

Specifies the intended recipient of the message

2.2.4.6 QString UdpMessage::Message

Message Data.

The actually string data contained in the message

2.2.4.7 `UdpMessageType` `UdpMessage::MessageType`

Message Type.

Specifies what type the message is.

2.2.4.8 `UdpUser*` `UdpMessage::SenderUser`

Sender User.

Specifies the user who send the message

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

- MessageAPI/udpobjects.h
- MessageAPI/udpobjects.cpp

2.3 `UdpUser` Class Reference

Public Member Functions

- `UdpUser` ()
UdpUser Constructor.
- `UdpUser` (QString)
UdpUser Constructor.
- `UdpUser` (QString, QString)
UdpUser Constructor.
- QString `getCurrentUser` ()
Get the Current User.

Public Attributes

- QString `Username`
Username.
- QString `Alias`
Alias.
- int `AdminLevel`
Admin Level.

Static Public Attributes

- static const QString [Anyone](#) = QString("ANY")
Anyone Username.
- static const QString [Self](#) = QString("SELF")
Self Username.

2.3.1 Constructor & Destructor Documentation

2.3.1.1 UdpUser::UdpUser ()

[UdpUser](#) Constructor.

Creates an empty [UdpUser](#)

2.3.1.2 UdpUser::UdpUser (QString *name*)

[UdpUser](#) Constructor.

Creates a [UdpUser](#) with a specified username

Parameters

<i>name</i>	the username to create the UdpUser with
-------------	---

2.3.1.3 UdpUser::UdpUser (QString *name*, QString *alias*)

[UdpUser](#) Constructor.

Creates a [UdpUser](#) which a specified username and alias

Parameters

<i>name</i>	the username to create the UdpUser with
<i>alias</i>	the alias to create the UdpUser with

2.3.2 Member Function Documentation

2.3.2.1 QString UdpUser::getCurrentUser ()

Get the Current User.

Gets the network-friendly username of the current logged in user.

Returns

the username of the current user

2.3.3 Member Data Documentation**2.3.3.1 int UdpUser::AdminLevel**

Admin Level.

The Admin Privilege Level of the User

2.3.3.2 QString UdpUser::Alias

Alias.

The alias of the User

2.3.3.3 const QString UdpUser::Anyone = QString("ANY") [static]

Anyone Username.

A special username which when specified sends a message everyone who is listening

2.3.3.4 const QString UdpUser::Self = QString("SELF") [static]

Self Username.

A special username which when specified sends a message only to the user who sent it.

2.3.3.5 QString UdpUser::Username

Username.

The network-friendly username of the User

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

- MessageAPI/udpobjects.h
- MessageAPI/udpobjects.cpp