Mobile Services
XMPP Chat supporting SIP

Christoph Kieslich, Hannes Markschläger
Manuel Lindorfer, Michael Schöllhammer

January 2013

1 Requirements

The goal of this project was to implement a XMPP chat which also supports the communication to SIP users. In order to provide this functionality there needs to be a conversion from either XMPP to SIP or SIP to XMPP messages. This conversion step is performed by a SIP-XMPP-Server which acts as a gateway between both protocols.

The following aspects have to be taken into account:

- **XMPP to XMPP:** Message exchange between two XMPP users
- **XMPP to SIP:** Message exchange between a XMPP and a SIP client
- **SIP to XMPP:** Message exchange between a SIP and a XMPP client

2 XMPP basics

The following chapter gibes an overview over the basics of XMPP, the communication based on XMPP and the addressing scheme.

2.1 What is XMPP?

XMPP stands for *Extensible Messaging and Presence Protocol* and is simply a message-oriented communication protocol based on XML. XMPP was developed by the *Jabber open-source community* in the late 90s for the main purpose of near-realtime instant messaging, including both presence information and contact list maintenance. Since it's development XMPP has been used for a lot of different applications, including publish-subscribe systems, signalling for VoIP, smart grids of social networking services, just to give a few examples.

The advantages XMPP offers can be summarized as following:

- **Decentralization:** The architecture of the XMPP network is similar to email; anyone can run their own XMPP server and there is no central master server.
- **Open standards:** The Internet Engineering Task Force has formalized XMPP as an approved instant messaging and presence technology under the name of XMPP. No royalties are required to implement support of these specifications and their development is not tied to a single vendor.
- **History:** Since XMPP technologies have been in use since 1999, multiple implementations of the XMPP standards exist for clients, servers, components, and code libraries.
- **Security:** XMPP servers can be isolated from the public XMPP network (e.g., on a company intranet), and strong security (via SASL and TLS) has been built into the core XMPP specifications.
- **Flexibility:** Custom functionality can be built on top of XMPP; to maintain interoperability, common extensions are managed by the XMPP Standards Foundation.

A drawback of XMPP on the the other side is that it's in-band binary data transfer is quite inefficient. Because XMPP is not yet encoded as Efficient XML Interchange but as a long-lived XML stream, binary data must be first base64 encoded before it can be transmitted in-band. Therefore any significant amount of binary data (eġ; file transfers) is best transmitted out-of-band, using in-band messages to coordinate.

2.2 XMPP Transport

The original and *native* transport protocol for XMPP is Transmission Control Protocol (TCP), using open-ended XML streams over long-lived TCP connections. As an alternative to the TCP transport, the XMPP community has also developed an HTTP transport for web clients as well as users behind restricted firewalls. In the original specification, XMPP could use HTTP in two ways: polling[16] and binding.

2.3 Decentralization and addressing

The XMPP network uses a client–server architecture (clients do not talk directly to one another). However, it is decentralized—by design, there is no central authoritative server, as there is with services such as AOL Instant Messenger or Windows Live Messenger. Some confusion often arises on this point as there is a public XMPP server being run at jabber.org, to which a large number of users subscribe. However, anyone may run their own XMPP server on their own domain.

Every user on the network has a unique ID. To avoid requiring a central server to maintain a list of IDs, the JID is structured like an email address with a username and a domain name (or IP address) for the server where that user resides, separated by an at sign (@), such as username@example.com.

3 Message exchange

The following chapter describes the basic principles of how the communication between different clients is ensured by either the clients or the SIP/XMPP-Server.

3.1 XMPP to XMPP

The client is able to send messages not only to SIP users via the gateway but also to normal XMPP users. As of now the client lacks the ability to add new contacts, therefore one has to use another client like *pidgin* to add XMPP users to his contact list. Once you have contacts in your list, those will appear in the XMPP client.

3.2 XMPP to SIP

In order to send a message from a XMPP to a SIP client, the message is not sent (as by default) to the twattle.net server, but instead to a interface at the SIP-XMPP-Server. This interface provides a method (see listing below) which simply extracts the neccessary information out of the XMPP message and forwards the message to the SIP destination.

Listing 1: Forwarding of XMPP Messages to SIP

```
/**
 * Forward message to SIP
 *
 * @param msg the msg
 */
private void forwardMessage(Message msg) {
  String to = m_relation_in.get(msg.getTo().split("@")[0]);
  Server.handleXMPPMessage(msg.getFrom(),to,msg.getBody());
}
```

3.3 SIP to XMPP

When a SIP client sends a message to a XMPP client, the SIP-XMPP-Server again extracts the neccessary information out of the SIP-message. A new XMPP message is created and forwarded to the corresponding SIP user over the twattle.net server. The listing below shows the information extraction and creation and forwarding of the new message.

Listing 2: Forwarding of XMPP Messages to SIP

```
/**
 * Handle sip message and forwards it to the xmpp user
 *
 * @param from the sip user
 * @param to the xmpp user
 * @param msg the message
 */
public void handleSIPMessage(String from, String to, String msg) {
  String fromUser =m_relation_out.get(from);
  Connection conn = m_connections.get(fromUser);
  try {
    to+="@twattle.net";//append domain
    Message message = new Message(to, Message.Type.chat);
    message.setBody(msg);
    conn.sendPacket(message);//send message to client
    m_logArea.append("message sent to: "+to);
  } catch (Exception e) {
    e.printStackTrace();
    m_logArea.append(e.getMessage());
  }
}
```

4 Components and technologies

The following itemization gives a short overview over the components and technologies used in our system.

- **Android XMPP Client**
- **twattle.net Server:** For handling the communication between two XMPP clients
- **XMPP-SIP-Gateway:** For handling the communication between either a XMPP and SIP or a SIP and XMPP client.
- **Used technologies**
  - XMPP as message protocol
  - Java smack library for the XMPP-SIP gateway
  - aSmack (Android) for the Android XMPP client

  i

5 Appendix

secureSMSAndroidClient

1.0

Generated by Doxygen 1.8.3.1

Mon Jan 21 2013 10:18:17

Contents

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

Chapter 2

Class Index

2.1 Class List

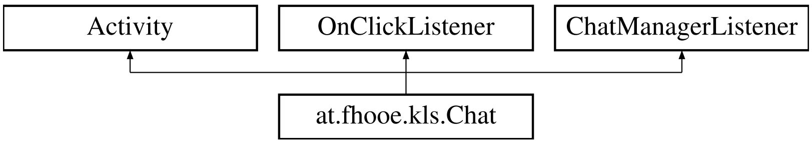
Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

Class Documentation

3.1 at.fhooe.kls.Chat Class Reference

Inheritance diagram for at.fhooe.kls.Chat:



Public Member Functions

- void onCreate (Bundle savedInstanceState)
- void **onClick** (View v)
- void setContact (ContactItem _contact)
- void **chatCreated** (org.jivesoftware.smack.Chat arg0, boolean arg1)

3.1.1 Detailed Description

The Class Chat.

Definition at line 27 of file Chat.java.

3.1.2 Member Function Documentation

3.1.2.1 void at.fhooe.kls.Chat.onCreate (Bundle *savedInstanceState*)

Called when the activity is first created.

Parameters

savedInstanceState	the saved instance state

Definition at line 51 of file Chat.java.

References at.fhooe.kls.XMPPClient.getConnection(), at.fhooe.kls.XMPPClient.getInstance(), and at.fhooe.kls.ContactItem.getUser().

3.1.2.2 void at.fhooe.kls.Chat.setContact (ContactItem *_contact*)

Sets the contact.

Parameters

_contact	the new contact

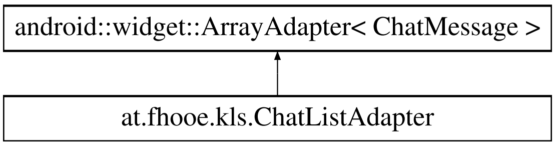
Definition at line 136 of file Chat.java.

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

- src/at/fhooe/kls/Chat.java

3.2 at.fhooe.kls.ChatListAdapter Class Reference

Inheritance diagram for at.fhooe.kls.ChatListAdapter:


android::widget::ArrayAdapter< ChatMessage >
at.fhooe.kls.ChatListAdapter

Public Member Functions

- ChatListAdapter (Context context, int textViewResourceId, ArrayList< ChatMessage > items)
- View **getView** (int position, View convertView, ViewGroup parent)

3.2.1 Detailed Description

The Class ChatListAdapter.

Definition at line 17 of file ChatListAdapter.java.

3.2.2 Constructor & Destructor Documentation

3.2.2.1 at.fhooe.kls.ChatListAdapter.ChatListAdapter (Context *context*, int *textViewResourceId*, ArrayList< ChatMessage > *items*)

Instantiates a new chat list adapter.

Parameters

context	the context
textViewResourceId	the text view resource id
items	the items

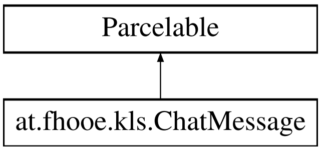
Definition at line 35 of file ChatListAdapter.java.

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

- src/at/fhooe/kls/ChatListAdapter.java

3.3 at.fhooe.kls.ChatMessage Class Reference

Inheritance diagram for at.fhooe.kls.ChatMessage:



Public Member Functions

- ChatMessage (String sender, Date time, String message)
- ChatMessage (String sender, Date time, String message, boolean mine)
- String getSender ()
- void setSender (String sender)
- Date getTime ()
- String shortTime ()
- void setTime (Date time)
- boolean isMine ()
- String getMessage ()
- void setMessage (String message)
- int **describeContents** ()
- void **writeToParcel** (Parcel dest, int flags)
- ChatMessage (Parcel in)

Static Public Attributes

- static final
  Parcelable.Creator
  < ChatMessage > CREATOR

3.3.1 Detailed Description

The Class ChatMessage.

Definition at line 12 of file ChatMessage.java.

3.3.2 Constructor & Destructor Documentation

3.3.2.1 at.fhooe.kls.ChatMessage.ChatMessage (String *sender,* Date *time,* String *message*)

Instantiates a new chat message.

Parameters

sender	the sender
time	the time
message	the message

Definition at line 33 of file ChatMessage.java.

3.3.2.2 at.fhooe.kls.ChatMessage.ChatMessage (String *sender,* Date *time,* String *message,* boolean *mine*)

Instantiates a new chat message.

Parameters

sender	the sender
time	the time
message	the message
mine	the mine

Definition at line 48 of file ChatMessage.java.

3.3.2.3 at.fhooe.kls.ChatMessage.ChatMessage (Parcel *in*)

Instantiates a new chat message.

Parameters

in	the in

Definition at line 167 of file ChatMessage.java.

3.3.3 Member Function Documentation

3.3.3.1 String at.fhooe.kls.ChatMessage.getMessage ( )

Gets the message.

Returns

the message

Definition at line 117 of file ChatMessage.java.

3.3.3.2 String at.fhooe.kls.ChatMessage.getSender ( )

Gets the sender.

Returns

the sender

Definition at line 58 of file ChatMessage.java.

3.3.3.3 Date at.fhooe.kls.ChatMessage.getTime ( )

Gets the time.

Returns

the time

Definition at line 76 of file ChatMessage.java.

3.3.3.4 boolean at.fhooe.kls.ChatMessage.isMine ( )

Checks if is mine.

Returns

true, if is mine

Definition at line 108 of file ChatMessage.java.

3.3.3.5 void at.fhooe.kls.ChatMessage.setMessage (String *message*)

Sets the message.

Parameters

message	the new message

Definition at line 126 of file ChatMessage.java.

3.3.3.6 void at.fhooe.kls.ChatMessage.setSender (String *sender*)

Sets the sender.

Parameters

sender	the new sender

Definition at line 67 of file ChatMessage.java.

3.3.3.7 void at.fhooe.kls.ChatMessage.setTime (Date *time*)

Sets the time.

Parameters

time	the new time

Definition at line 99 of file ChatMessage.java.

3.3.3.8 String at.fhooe.kls.ChatMessage.shortTime ( )

Short time.

Returns

the string

Definition at line 85 of file ChatMessage.java.

3.3.4 Member Data Documentation

3.3.4.1 final Parcelable.Creator<ChatMessage> at.fhooe.kls.ChatMessage.CREATOR `[static]`

Initial value:

```
= new Parcelable.Creator<ChatMessage>() {
        public ChatMessage createFromParcel(Parcel in) {
            return new ChatMessage(in);
        }

        public ChatMessage[] newArray(int size) {
            return new ChatMessage[size];
        }
    }
```

The Constant CREATOR.

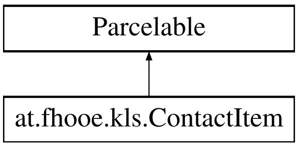
Definition at line 152 of file ChatMessage.java.

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

- src/at/fhooe/kls/ChatMessage.java

3.4 at.fhooe.kls.ContactItem Class Reference

Inheritance diagram for at.fhooe.kls.ContactItem:



Public Member Functions

- String getStatus ()
- void setStatus (String status)
- String getKey ()
- void setKey (String key)
- ContactItem (String user)
- boolean isAvailable ()
- void setAvailable (boolean available)
- String getUser ()
- void setUser (String user)
- int **describeContents** ()
- void **writeToParcel** (Parcel dest, int flags)

Static Public Attributes

- static final
  Parcelable.Creator
  < ContactItem > CREATOR

3.4.1 Detailed Description

The Class ContactItem.

Definition at line 10 of file ContactItem.java.

3.4.2 Constructor & Destructor Documentation

3.4.2.1 at.fhooe.kls.ContactItem.ContactItem (String *user*)

Instantiates a new contact item.

Parameters

user	the user

Definition at line 69 of file ContactItem.java.

3.4.3 Member Function Documentation

3.4.3.1 String at.fhooe.kls.ContactItem.getKey ( )

Gets the key.

Returns

the key

Definition at line 49 of file ContactItem.java.

3.4.3.2 String at.fhooe.kls.ContactItem.getStatus ( )

Gets the status.

Returns

the status

Definition at line 29 of file ContactItem.java.

3.4.3.3 String at.fhooe.kls.ContactItem.getUser ( )

Gets the user.

Returns

the user

Definition at line 101 of file ContactItem.java.

Referenced by at.fhooe.kls.Chat.onCreate().

3.4.3.4 boolean at.fhooe.kls.ContactItem.isAvailable ( )

Checks if is available.

Returns

true, if is available

Definition at line 81 of file ContactItem.java.

3.4.3.5 void at.fhooe.kls.ContactItem.setAvailable (boolean *available*)

Sets the available.

Parameters

available	the new available

Definition at line 91 of file ContactItem.java.

Referenced by at.fhooe.kls.SecureSMSActivity.onCreate().

3.4.3.6 void at.fhooe.kls.ContactItem.setKey (String *key*)

Sets the key.

Parameters

key	the new key

Definition at line 59 of file ContactItem.java.

3.4.3.7 void at.fhooe.kls.ContactItem.setStatus (String *status*)

Sets the status.

Parameters

status	the new status

Definition at line 39 of file ContactItem.java.

Referenced by at.fhooe.kls.SecureSMSActivity.onCreate().

3.4.3.8 void at.fhooe.kls.ContactItem.setUser (String *user*)

Sets the user.

Parameters

user	the new user

Definition at line 110 of file ContactItem.java.

3.4.4 Member Data Documentation

3.4.4.1 final Parcelable.Creator<ContactItem> at.fhooe.kls.ContactItem.CREATOR [static]

Initial value:

```
= new Parcelable.Creator<ContactItem>() {
        public ContactItem createFromParcel(Parcel in) {
            return new ContactItem(in);
        }

        public ContactItem[] newArray(int size) {
            return new ContactItem[size];
        }
    }
```

The Constant CREATOR.

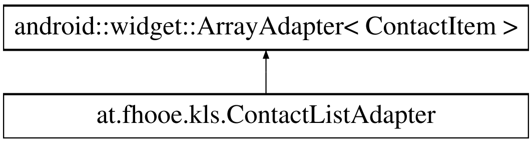
Definition at line 124 of file ContactItem.java.

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

- src/at/fhooe/kls/ContactItem.java

3.5 at.fhooe.kls.ContactListAdapter Class Reference

Inheritance diagram for at.fhooe.kls.ContactListAdapter:


android::widget::ArrayAdapter< ContactItem >
at.fhooe.kls.ContactListAdapter

Public Member Functions

- ContactListAdapter (Context context, int textViewResourceId, ArrayList< ContactItem > items)
- View **getView** (int position, View convertView, ViewGroup parent)

3.5.1 Detailed Description

The Class ContactListAdapter.

Definition at line 16 of file ContactListAdapter.java.

3.5.2 Constructor & Destructor Documentation

3.5.2.1 at.fhooe.kls.ContactListAdapter.ContactListAdapter (Context *context,* int *textViewResourceId,* ArrayList< ContactItem > *items*)

Instantiates a new contact list adapter.

Parameters

context	the context
textViewResourceId	the text view resource id
items	the items

Definition at line 34 of file ContactListAdapter.java.

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

- src/at/fhooe/kls/ContactListAdapter.java

3.6 at.fhooe.kls.Hash Class Reference

Static Public Member Functions

- static byte[] SHA256 (byte[] text) throws NoSuchAlgorithmException

Static Public Attributes

- static final String DIGEST_ALGORITHM = "SHA-256"

3.6.1 Detailed Description

The Class Hash.

Definition at line 9 of file Hash.java.

3.6.2 Member Function Documentation

3.6.2.1 static byte [ ] at.fhooe.kls.Hash.SHA256 (byte[ ] *text*) throws NoSuchAlgorithmException [static]

Generates a SHA256.

Parameters

text	plaintext

Returns

the byte[] hashed plaintext

Exceptions

NoSuchAlgorithmException	the no such algorithm exception

Definition at line 20 of file Hash.java.

References at.fhooe.kls.Hash.DIGEST_ALGORITHM.

Referenced by at.fhooe.kls.SMSAdapter.createNote(), at.fhooe.kls.SMSAdapter.getKey(), and at.fhooe.kls.SMSAdapter.open().

3.6.3 Member Data Documentation

3.6.3.1 final String at.fhooe.kls.Hash.DIGEST_ALGORITHM = "SHA-256" [static]

The Constant DIGEST_ALGORITHM.

Definition at line 11 of file Hash.java.

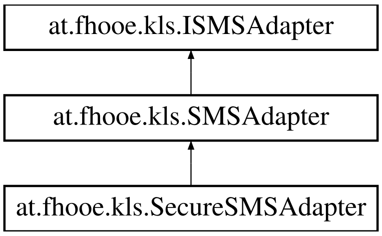
Referenced by at.fhooe.kls.Hash.SHA256().

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

- src/at/fhooe/kls/Hash.java

3.7 at.fhooe.kls.ISMSAdapter Interface Reference

Inheritance diagram for at.fhooe.kls.ISMSAdapter:



Public Member Functions

- boolean isOpen ()
- void close ()
- long createNote (String email, String key)
- String getKey (String email)
- Cursor fetchNote (String email) throws SQLException
- SMSAdapter open (String password) throws SQLException

3.7.1 Detailed Description

The Interface ISMSAdapter.

Definition at line 11 of file ISMSAdapter.java.

3.7.2 Member Function Documentation

3.7.2.1 void at.fhooe.kls.ISMSAdapter.close ( )

Close.

Implemented in at.fhooe.kls.SMSAdapter, and at.fhooe.kls.SecureSMSAdapter.

3.7.2.2 long at.fhooe.kls.ISMSAdapter.createNote (String *email*, String *key*)

Creates the note.

Parameters

email	the email
key	the key

Returns

the long

Implemented in at.fhooe.kls.SMSAdapter, and at.fhooe.kls.SecureSMSAdapter.

3.7.2.3 Cursor at.fhooe.kls.ISMSAdapter.fetchNote (String *email*) throws SQLException

Fetch note.

Parameters

email	the email

Returns

the cursor

Exceptions

SQLException	the sQL exception

Implemented in at.fhooe.kls.SMSAdapter, and at.fhooe.kls.SecureSMSAdapter.

3.7.2.4 String at.fhooe.kls.ISMSAdapter.getKey (String *email*)

Gets the key.

Parameters

email	the email

Returns

the key

Implemented in at.fhooe.kls.SMSAdapter, and at.fhooe.kls.SecureSMSAdapter.

3.7.2.5 boolean at.fhooe.kls.ISMSAdapter.isOpen ( )

Checks if is open.

Returns

true, if is open

Implemented in at.fhooe.kls.SMSAdapter, and at.fhooe.kls.SecureSMSAdapter.

3.7.2.6 SMSAdapter at.fhooe.kls.ISMSAdapter.open (String *password*) throws SQLException

Open.

Parameters

password	the password

Returns

the sMS adapter

Exceptions

SQLException	the sQL exception

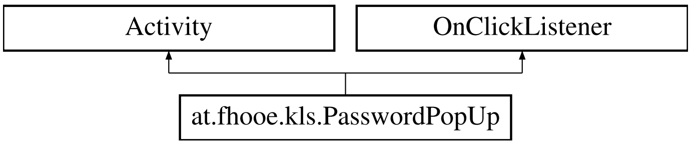
Implemented in at.fhooe.kls.SMSAdapter, and at.fhooe.kls.SecureSMSAdapter.

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

- src/at/fhooe/kls/ISMSAdapter.java

3.8 at.fhooe.kls.PasswordPopUp Class Reference

Inheritance diagram for at.fhooe.kls.PasswordPopUp:



Public Member Functions

- void **onClick** (View view)

Protected Member Functions

- void **onCreate** (Bundle savedInstanceState)
- void **onStart** ()
- void **onResume** ()
- void **onPause** ()
- void **onStop** ()
- void **onDestroy** ()

3.8.1 Detailed Description

The Class PasswordPopUp.

Definition at line 18 of file PasswordPopUp.java.

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

- src/at/fhooe/kls/PasswordPopUp.java

3.9 at.fhooe.kls.SecureSharedPrefAdapter Class Reference

Static Public Member Functions

- static boolean **open** (Context _ctx, String _password)
- static String **getKey** (String email)
- static void **insertKey** (String email, String key)

3.9.1 Detailed Description

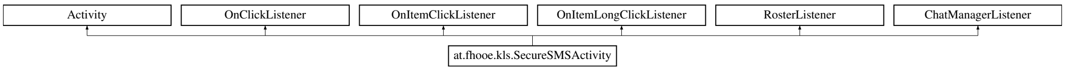
Definition at line 11 of file SecureSharedPrefAdapter.java.

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

- src/at/fhooe/kls/SecureSharedPrefAdapter.java

3.10 at.fhooe.kls.SecureSMSActivity Class Reference

Inheritance diagram for at.fhooe.kls.SecureSMSActivity:



Public Member Functions

- void onCreate (Bundle savedInstanceState)
- void **onActivityResult** (int requestCode, int resultCode, Intent intent)
- void **onClick** (View v)
- void showQrCode (String data)
- void **onItemClick** (AdapterView<?> parent, View v, int pos, long arg3)
- boolean **onItemLongClick** (AdapterView<?> parent, View v, int pos, long arg3)
- void **entriesAdded** (Collection< String > addresses)
- void **entriesDeleted** (Collection< String > addresses)
- void **entriesUpdated** (Collection< String > addresses)
- void **presenceChanged** (Presence presence)
- void **chatCreated** (org.jivesoftware.smack.Chat chat, boolean createdLocally)

Protected Member Functions

- void **onDestroy** ()
- void **onResume** ()
- void **onStart** ()
- void **onPause** ()

3.10.1 Detailed Description

The Class SecureSMSActivity.

Definition at line 40 of file SecureSMSActivity.java.

3.10.2 Member Function Documentation

3.10.2.1 void at.fhooe.kls.SecureSMSActivity.onCreate (Bundle *savedInstanceState*)

Called when the activity is first created.

Parameters

savedInstanceState	the saved instance state

Definition at line 78 of file SecureSMSActivity.java.

References at.fhooe.kls.XMPPClient.getInstance(), at.fhooe.kls.XMPPClient.getRoster(), at.fhooe.kls.XMPPClient.isConnected(), at.fhooe.kls.ContactItem.setAvailable(), at.fhooe.kls.ContactItem.setStatus(), and at.fhooe.kls.Util.showToast().

3.10.2.2 void at.fhooe.kls.SecureSMSActivity.showQrCode (String *data*)

Show qr code.

Parameters

data	the data

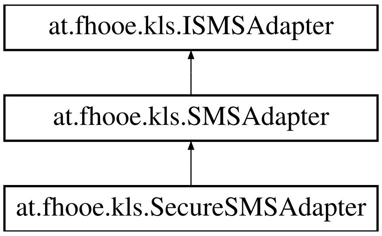
Definition at line 256 of file SecureSMSActivity.java.

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

- src/at/fhooe/kls/SecureSMSActivity.java

3.11 at.fhooe.kls.SecureSMSAdapter Class Reference

Inheritance diagram for at.fhooe.kls.SecureSMSAdapter:



Public Member Functions

- SecureSMSAdapter open (String password) throws SQLException
- boolean isOpen ()
- void close ()
- long createNote (String email, String key)
- Cursor fetchNote (String email) throws SQLException
- String getKey (String email)

Static Public Member Functions

- static synchronized
  SecureSMSAdapter getInstance (Context ctx)

Additional Inherited Members

3.11.1 Detailed Description

The Class SecureSMSAdapter.

Definition at line 15 of file SecureSMSAdapter.java.

3.11.2 Member Function Documentation

3.11.2.1 void at.fhooe.kls.SecureSMSAdapter.close ( )

Close.

Implements at.fhooe.kls.ISMSAdapter.

Definition at line 135 of file SecureSMSAdapter.java.

3.11.2.2 long at.fhooe.kls.SecureSMSAdapter.createNote (String *email,* String *key*)

Creates the note.

Parameters

email	the email
key	the key

Returns

the long

Implements at.fhooe.kls.ISMSAdapter.

Definition at line 143 of file SecureSMSAdapter.java.

3.11.2.3 Cursor at.fhooe.kls.SecureSMSAdapter.fetchNote (String *email*) throws SQLException

Fetch note.

Parameters

email	the email

Returns

the cursor

Exceptions

SQLException	the sQL exception

Implements at.fhooe.kls.ISMSAdapter.

Definition at line 157 of file SecureSMSAdapter.java.

Referenced by at.fhooe.kls.SecureSMSAdapter.getKey().

3.11.2.4 static synchronized SecureSMSAdapter at.fhooe.kls.SecureSMSAdapter.getInstance (Context *ctx*) [static]

Gets the single instance of SecureSMSAdapter.

Parameters

ctx	the ctx

Returns

single instance of SecureSMSAdapter

Definition at line 100 of file SecureSMSAdapter.java.

3.11.2.5 String at.fhooe.kls.SecureSMSAdapter.getKey (String *email*)

Gets the key.

Parameters

email	the email

Returns

the key

Implements at.fhooe.kls.ISMSAdapter.

Definition at line 174 of file SecureSMSAdapter.java.

References at.fhooe.kls.SecureSMSAdapter.fetchNote().

3.11.2.6 boolean at.fhooe.kls.SecureSMSAdapter.isOpen ( )

Checks if is open.

Returns

true, if is open

Implements at.fhooe.kls.ISMSAdapter.

Definition at line 125 of file SecureSMSAdapter.java.

3.11.2.7 SecureSMSAdapter at.fhooe.kls.SecureSMSAdapter.open (String *password*) throws SQLException

Open.

Parameters

password	the password

Returns

the sMS adapter

Exceptions

SQLException	the sQL exception

Implements at.fhooe.kls.ISMSAdapter.

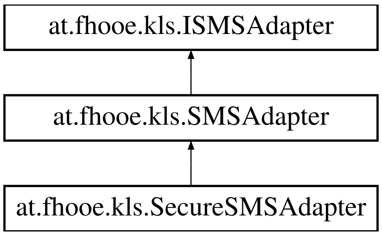
Definition at line 113 of file SecureSMSAdapter.java.

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

- src/at/fhooe/kls/SecureSMSAdapter.java

3.12 at.fhooe.kls.SMSAdapter Class Reference

Inheritance diagram for at.fhooe.kls.SMSAdapter:



Public Member Functions

- SMSAdapter open (String password) throws SQLException
- boolean isOpen ()
- void close ()
- long createNote (String email, String key)
- String getKey (String email)
- Cursor fetchNote (String email) throws SQLException

Static Public Member Functions

- static synchronized SMSAdapter getInstance (Context ctx)

Static Public Attributes

- static final String PREFS_NAME = "smsadapter"

Protected Member Functions

- SMSAdapter (Context ctx)

3.12.1 Detailed Description

The Class SMSAdapter.

Definition at line 18 of file SMSAdapter.java.

3.12.2 Constructor & Destructor Documentation

3.12.2.1 at.fhooe.kls.SMSAdapter.SMSAdapter (Context *ctx*) [protected]

Instantiates a new sMS adapter.

Parameters

ctx	the ctx

Definition at line 109 of file SMSAdapter.java.

Referenced by at.fhooe.kls.SMSAdapter.getInstance().

3.12.3 Member Function Documentation

3.12.3.1 void at.fhooe.kls.SMSAdapter.close ( )

Close.

Implements at.fhooe.kls.ISMSAdapter.

Definition at line 185 of file SMSAdapter.java.

3.12.3.2 long at.fhooe.kls.SMSAdapter.createNote (String *email*, String *key*)

Creates the note.

Parameters

email	the email
key	the key

Returns

the long

Implements at.fhooe.kls.ISMSAdapter.

Definition at line 196 of file SMSAdapter.java.

References at.fhooe.kls.Util.encrypt(), at.fhooe.kls.SMSAdapter.getKey(), at.fhooe.kls.Hash.SHA256(), and at.fhooe.kls.Util.toHex().

3.12.3.3 Cursor at.fhooe.kls.SMSAdapter.fetchNote (String *email*) throws SQLException

Fetch note.

Parameters

email	the email

Returns

the cursor

Exceptions

SQLException	the sQL exception

Implements at.fhooe.kls.ISMSAdapter.

Definition at line 257 of file SMSAdapter.java.

Referenced by at.fhooe.kls.SMSAdapter.getKey().

3.12.3.4 static synchronized SMSAdapter at.fhooe.kls.SMSAdapter.getInstance (Context *ctx*) [static]

Gets the single instance of SMSAdapter.

Parameters

ctx	the ctx

Returns

single instance of SMSAdapter

Definition at line 122 of file SMSAdapter.java.

References at.fhooe.kls.SMSAdapter.SMSAdapter().

3.12.3.5 String at.fhooe.kls.SMSAdapter.getKey (String *email*)

Gets the key.

Parameters

email	the email

Returns

the key

Implements at.fhooe.kls.ISMSAdapter.

Definition at line 232 of file SMSAdapter.java.

References at.fhooe.kls.Util.decrypt(), at.fhooe.kls.SMSAdapter.fetchNote(), at.fhooe.kls.Hash.SHA256(), and at.fhooe.kls.Util.toByte().

Referenced by at.fhooe.kls.SMSAdapter.createNote().

3.12.3.6 boolean at.fhooe.kls.SMSAdapter.isOpen ( )

Checks if is open.

Returns

true, if is open

Implements at.fhooe.kls.ISMSAdapter.

Definition at line 173 of file SMSAdapter.java.

3.12.3.7 SMSAdapter at.fhooe.kls.SMSAdapter.open (String *password*) throws SQLException

Open.

Parameters

password	the password

Returns

the sMS adapter

Exceptions

SQLException	the sQL exception

Implements at.fhooe.kls.ISMSAdapter.

Definition at line 137 of file SMSAdapter.java.

References at.fhooe.kls.SMSAdapter.PREFS_NAME, at.fhooe.kls.Hash.SHA256(), and at.fhooe.kls.Util.toHex().

3.12.4 Member Data Documentation

3.12.4.1 final String at.fhooe.kls.SMSAdapter.PREFS_NAME = "smsadapter" [static]

The Constant PREFS_NAME.

Definition at line 21 of file SMSAdapter.java.

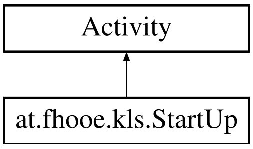
Referenced by at.fhooe.kls.SMSAdapter.open().

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

- src/at/fhooe/kls/SMSAdapter.java

3.13 at.fhooe.kls.StartUp Class Reference

Inheritance diagram for at.fhooe.kls.StartUp:



Classes

- enum **States**

Static Public Attributes

- static final String PREFS_NAME = "MyPrefsFile"

Protected Member Functions

- void **onCreate** (Bundle savedInstanceState)
- void checkLogin (String username, String password)

3.13.1 Detailed Description

The Class StartUp.

Definition at line 28 of file StartUp.java.

3.13.2 Member Function Documentation

3.13.2.1 void at.fhooe.kls.StartUp.checkLogin (String *username,* String *password*) [protected]

Check login.

Parameters

username	the username
password	the password

Definition at line 173 of file StartUp.java.

3.13.3 Member Data Documentation

3.13.3.1 final String at.fhooe.kls.StartUp.PREFS_NAME = "MyPrefsFile" [static]

The Constant PREFS_NAME.

Definition at line 53 of file StartUp.java.

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

- src/at/fhooe/kls/StartUp.java

3.14 at.fhooe.kls.Util Class Reference

Static Public Member Functions

- static String toHex (byte[] buf)
- static byte[] getRawKey (byte[] seed) throws Exception
- static void showToast (Context context, String message)
- static byte[] encrypt (byte[] raw, byte[] clear) throws Exception
- static byte[] decrypt (byte[] raw, byte[] encrypted) throws Exception
- static byte[] toByte (String hexString)
- static byte[] generateSeed ()
- static String getKey () throws Exception

3.14.1 Detailed Description

The Class Util.

Definition at line 21 of file Util.java.

3.14.2 Member Function Documentation

3.14.2.1 static byte [] at.fhooe.kls.Util.decrypt (byte[] *raw*, byte[] *encrypted*) throws Exception [static]

Decrypt.

Parameters

raw	the raw
encrypted	the encrypted

Returns

the byte[]

Exceptions

Exception	the exception

Definition at line 108 of file Util.java.

Referenced by at.fhooe.kls.SMSAdapter.getKey().

3.14.2.2 static byte [ ] at.fhooe.kls.Util.encrypt (byte[ ] *raw,* byte[ ] *clear*) throws Exception [static]

Encrypt.

Parameters

raw	the raw
clear	the clear

Returns

the byte[]

Exceptions

Exception	the exception

Definition at line 92 of file Util.java.

Referenced by at.fhooe.kls.SMSAdapter.createNote().

3.14.2.3 static byte [ ] at.fhooe.kls.Util.generateSeed ( ) [static]

Generate seed.

Returns

the byte[]

Definition at line 135 of file Util.java.

Referenced by at.fhooe.kls.Util.getKey().

3.14.2.4 static String at.fhooe.kls.Util.getKey ( ) throws Exception [static]

Gets the key.

Returns

the key

Exceptions

Exception	the exception

Definition at line 170 of file Util.java.

References at.fhooe.kls.Util.generateSeed(), at.fhooe.kls.Util.getRawKey(), and at.fhooe.kls.Util.toHex().

3.14.2.5 static byte [ ] at.fhooe.kls.Util.getRawKey (byte[ ] *seed*) throws Exception [static]

Gets the raw key.

Parameters

seed	the seed

Returns

the raw key

Exceptions

Exception	the exception

Definition at line 62 of file Util.java.

Referenced by at.fhooe.kls.Util.getKey().

3.14.2.6 static void at.fhooe.kls.Util.showToast (Context *context,* String *message*) [static]

Show toast.

Parameters

context	the context
message	the message

Definition at line 78 of file Util.java.

Referenced by at.fhooe.kls.SecureSMSActivity.onCreate().

3.14.2.7 static byte [ ] at.fhooe.kls.Util.toByte (String *hexString*) [static]

To byte.

Parameters

hexString	the hex string

Returns

the byte[]

Definition at line 122 of file Util.java.

Referenced by at.fhooe.kls.SMSAdapter.getKey().

3.14.2.8 static String at.fhooe.kls.Util.toHex (byte[ ] *buf*) [static]

To hex.

Parameters

buf	the buf

Returns

the string

Definition at line 32 of file Util.java.

Referenced by at.fhooe.kls.SMSAdapter.createNote(), at.fhooe.kls.Util.getKey(), and at.fhooe.kls.SMSAdapter.-open().

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

- src/at/fhooe/kls/Util.java

3.15 at.fhooe.kls.XMPPClient Class Reference

Public Member Functions

- boolean login (String _user, String _pass)
- synchronized XMPPConnection getConnection ()
- void sendMessage (String to, String text)
- Roster getRoster ()
- boolean isConnected ()

Static Public Member Functions

- static synchronized XMPPClient getInstance ()

3.15.1 Detailed Description

The Class XMPPClient.

Definition at line 15 of file XMPPClient.java.

3.15.2 Member Function Documentation

3.15.2.1 synchronized XMPPConnection at.fhooe.kls.XMPPClient.getConnection ( )

Gets the connection.

Returns

the connection

Definition at line 59 of file XMPPClient.java.

Referenced by at.fhooe.kls.Chat.onCreate().

3.15.2.2 static synchronized XMPPClient at.fhooe.kls.XMPPClient.getInstance ( ) [static]

Gets the single instance of XMPPClient.

Returns

single instance of XMPPClient

Definition at line 69 of file XMPPClient.java.

Referenced by at.fhooe.kls.Chat.onCreate(), and at.fhooe.kls.SecureSMSActivity.onCreate().

3.15.2.3 Roster at.fhooe.kls.XMPPClient.getRoster ( )

Gets the roster.

Returns

the roster

Definition at line 99 of file XMPPClient.java.

Referenced by at.fhooe.kls.SecureSMSActivity.onCreate().

3.15.2.4 boolean at.fhooe.kls.XMPPClient.isConnected ( )

Checks if is connected.

Returns

true, if is connected

Definition at line 108 of file XMPPClient.java.

Referenced by at.fhooe.kls.SecureSMSActivity.onCreate().

3.15.2.5 boolean at.fhooe.kls.XMPPClient.login (String *_user,* String *_pass*)

Login.

Parameters

_user	the _user
_pass	the _pass

Returns

true, if successful

Definition at line 30 of file XMPPClient.java.

References at.fhooe.kls.XMPPLogic.getInstance(), and at.fhooe.kls.XMPPLogic.setConnection().

3.15.2.6 void at.fhooe.kls.XMPPClient.sendMessage (String *to,* String *text*)

Send message.

Parameters

to	the to
text	the text

Definition at line 82 of file XMPPClient.java.

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

- src/at/fhooe/kls/XMPPClient.java

3.16 at.fhooe.kls.XMPPLogic Class Reference

Public Member Functions

- void setConnection (XMPPConnection connection)
- XMPPConnection getConnection ()

Static Public Member Functions

- static synchronized XMPPLogic getInstance ()

3.16.1 Detailed Description

The Class XMPPLogic.

Definition at line 10 of file XMPPLogic.java.

3.16.2 Member Function Documentation

3.16.2.1 XMPPConnection at.fhooe.kls.XMPPLogic.getConnection ( )

Gets the connection.

Returns

the connection

Definition at line 44 of file XMPPLogic.java.

3.16.2.2 static synchronized XMPPLogic at.fhooe.kls.XMPPLogic.getInstance ( ) [static]

Gets the single instance of XMPPLogic.

Returns

single instance of XMPPLogic

Definition at line 23 of file XMPPLogic.java.

Referenced by at.fhooe.kls.XMPPClient.login().

3.16.2.3 void at.fhooe.kls.XMPPLogic.setConnection (XMPPConnection *connection*)

Sets the connection.

Parameters

connection	the new connection

Definition at line 35 of file XMPPLogic.java.

Referenced by at.fhooe.kls.XMPPClient.login().

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

- src/at/fhooe/kls/XMPPLogic.java

Index

XMPPGateway

Generated by Doxygen 1.8.3.1

Mon Jan 21 2013 10:18:01

Contents

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

Chapter 4

Class Documentation

4.1 Server Class Reference

Static Public Member Functions

- static void handleXMPPMessage (String from, String to, String body)

4.1.1 Detailed Description

Definition at line 2 of file Server.java.

4.1.2 Member Function Documentation

4.1.2.1 static void Server.handleXMPPMessage (String *from*, String *to*, String *body*) [static]

Definition at line 6 of file Server.java.

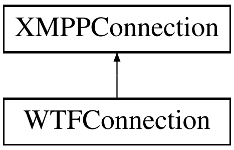
Referenced by XmppPanel.forwardMessage().

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

- src/Server.java

4.2 WTFConnection Class Reference

Inheritance diagram for WTFConnection:



Public Member Functions

- WTFConnection (ConnectionConfiguration config)
- WTFConnection (String server)
- WTFConnection ()

- String toString ()

4.2.1 Detailed Description

Definition at line 5 of file WTFConnection.java.

4.2.2 Constructor & Destructor Documentation

4.2.2.1 WTFConnection.WTFConnection (ConnectionConfiguration *config*)

Definition at line 7 of file WTFConnection.java.

4.2.2.2 WTFConnection.WTFConnection (String *server*)

Definition at line 11 of file WTFConnection.java.

4.2.2.3 WTFConnection.WTFConnection ( )

Definition at line 17 of file WTFConnection.java.

4.2.3 Member Function Documentation

4.2.3.1 String WTFConnection.toString ( )

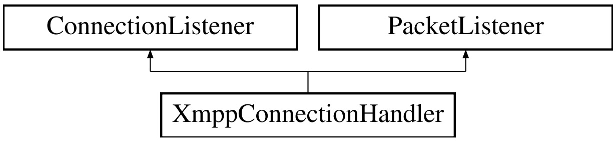
Definition at line 24 of file WTFConnection.java.

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

- src/WTFConnection.java

4.3 XmppConnectionHandler Class Reference

Inheritance diagram for XmppConnectionHandler:



Public Member Functions

- XmppConnectionHandler (Connection _conn)
- void connectionClosed ()
- void connectionClosedOnError (Exception _e)
- void reconnectingIn (int arg0)
- void reconnectionFailed (Exception arg0)
- void reconnectionSuccessful ()
- void processPacket (Packet _packet)
- void printPacket (Packet _packet)

4.3.1 Detailed Description

Definition at line 12 of file XmppConnectionHandler.java.

4.3.2 Constructor & Destructor Documentation

4.3.2.1 XmppConnectionHandler.XmppConnectionHandler (Connection *_conn*)

Definition at line 14 of file XmppConnectionHandler.java.

4.3.3 Member Function Documentation

4.3.3.1 void XmppConnectionHandler.connectionClosed ( )

Definition at line 24 of file XmppConnectionHandler.java.

4.3.3.2 void XmppConnectionHandler.connectionClosedOnError (Exception *_e*)

Definition at line 30 of file XmppConnectionHandler.java.

4.3.3.3 void XmppConnectionHandler.printPacket (Packet *_packet*)

Definition at line 58 of file XmppConnectionHandler.java.

Referenced by processPacket().

4.3.3.4 void XmppConnectionHandler.processPacket (Packet *_packet*)

Definition at line 54 of file XmppConnectionHandler.java.

References printPacket().

4.3.3.5 void XmppConnectionHandler.reconnectingIn (int *arg0*)

Definition at line 36 of file XmppConnectionHandler.java.

4.3.3.6 void XmppConnectionHandler.reconnectionFailed (Exception *arg0*)

Definition at line 42 of file XmppConnectionHandler.java.

4.3.3.7 void XmppConnectionHandler.reconnectionSuccessful ( )

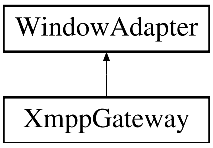
Definition at line 48 of file XmppConnectionHandler.java.

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

- src/XmppConnectionHandler.java

4.4 XmppGateway Class Reference

Inheritance diagram for XmppGateway:



Public Member Functions

- XmppGateway ()
- void initUI ()
- void windowClosing (WindowEvent e)

Static Public Member Functions

- static void main (String[ ] args)

4.4.1 Detailed Description

Definition at line 15 of file XmppGateway.java.

4.4.2 Constructor & Destructor Documentation

4.4.2.1 XmppGateway.XmppGateway ( )

Definition at line 18 of file XmppGateway.java.

References initUI().

Referenced by main().

4.4.3 Member Function Documentation

4.4.3.1 void XmppGateway.initUI ( )

Definition at line 23 of file XmppGateway.java.

Referenced by XmppGateway().

4.4.3.2 static void XmppGateway.main (String[ ] *args*) `[static]`

Parameters

args	

Definition at line 40 of file XmppGateway.java.

References XmppGateway().

4.4.3.3 void XmppGateway.windowClosing (WindowEvent *e*)

Definition at line 50 of file XmppGateway.java.

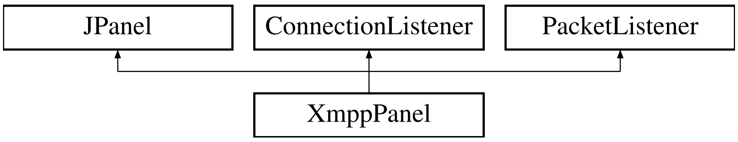
References XmppPanel.disconnect().

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

- src/XmppGateway.java

4.5 XmppPanel Class Reference

Inheritance diagram for XmppPanel:



Public Member Functions

- XmppPanel ()
- void connect ()
- void disconnect ()
- void processPacket (Packet _packet)
- void printMessage (Message msg)
- void handleSIPMessage (String from, String to, String msg)
- void connectionClosed ()
- void connectionClosedOnError (Exception e)
- void reconnectingIn (int arg0)
- void reconnectionFailed (Exception arg0)
- void reconnectionSuccessful ()

Private Member Functions

- void forwardMessage (Message msg)

Private Attributes

- HashMap< String, WTFConnection > m_connections = new HashMap<>()
- HashMap< String, String > m_accounts = new HashMap<>()
- HashMap< String, String > m_relation_in = new HashMap<>()
- HashMap< String, String > m_relation_out = new HashMap<>()
- JList m_connectionList
- JTextArea m_logArea
- JScrollPane scrollPane_2
- DefaultListModel< WTFConnection > m_listModel
- JLabel lblXmppConnections
- JLabel label
- JLabel label_1
- ConnectionConfiguration xmppConfig
- ConnectionConfiguration m_config

Static Private Attributes

- static final long serialVersionUID = 5657339239939443076L
- static final int DELAY = 1000

4.5.1 Detailed Description

The Class XmppPanel.

Definition at line 29 of file XmppPanel.java.

4.5.2 Constructor & Destructor Documentation

4.5.2.1 XmppPanel.XmppPanel ( )

Instantiates a new xmpp panel.

Definition at line 80 of file XmppPanel.java.

References connect(), label, label_1, lblXmppConnections, m_accounts, m_config, m_connectionList, m_listModel, m_logArea, m_relation_in, m_relation_out, and scrollPane_2.

4.5.3 Member Function Documentation

4.5.3.1 void XmppPanel.connect ( )

Connect all the gateway clients

Definition at line 140 of file XmppPanel.java.

References DELAY, m_accounts, m_config, m_connections, and m_listModel.

Referenced by XmppPanel().

4.5.3.2 void XmppPanel.connectionClosed ( )

Definition at line 283 of file XmppPanel.java.

4.5.3.3 void XmppPanel.connectionClosedOnError (Exception *e*)

Definition at line 291 of file XmppPanel.java.

4.5.3.4 void XmppPanel.disconnect ( )

Disconnect the gateway clients

Definition at line 190 of file XmppPanel.java.

References m_connections.

Referenced by XmppGateway.windowClosing().

4.5.3.5 void XmppPanel.forwardMessage (Message *msg*) `[private]`

Forward message to SIP

Parameters

msg	the msg

Definition at line 223 of file XmppPanel.java.

References Server.handleXMPPMessage(), and m_relation_in.

Referenced by processPacket().

4.5.3.6 void XmppPanel.handleSIPMessage (String *from,* String *to,* String *msg*)

Handle sip message and forwards it to the xmpp user

Parameters

from	the sip user
to	the xmpp user
msg	the message

Definition at line 255 of file XmppPanel.java.

References m_connections, m_logArea, and m_relation_out.

4.5.3.7 void XmppPanel.printMessage (Message *msg*)

Prints the Message on the screen and on the standard output

Parameters

msg	the Message which should be printed

Definition at line 234 of file XmppPanel.java.

References m_logArea, and m_relation_in.

Referenced by processPacket().

4.5.3.8 void XmppPanel.processPacket (Packet *_packet*)

Definition at line 205 of file XmppPanel.java.

References forwardMessage(), and printMessage().

4.5.3.9 void XmppPanel.reconnectingIn (int *arg0*)

Definition at line 299 of file XmppPanel.java.

4.5.3.10 void XmppPanel.reconnectionFailed (Exception *arg0*)

Definition at line 307 of file XmppPanel.java.

4.5.3.11 void XmppPanel.reconnectionSuccessful ( )

Definition at line 315 of file XmppPanel.java.

4.5.4 Member Data Documentation

4.5.4.1 final int XmppPanel.DELAY = 1000 [static], [private]

The Constant DELAY.

Definition at line 75 of file XmppPanel.java.

Referenced by connect().

4.5.4.2 JLabel XmppPanel.label [private]

The label.

Definition at line 63 of file XmppPanel.java.

Referenced by XmppPanel().

4.5.4.3 JLabel XmppPanel.label_1 [private]

The label_1.

Definition at line 66 of file XmppPanel.java.

Referenced by XmppPanel().

4.5.4.4 JLabel XmppPanel.lblXmppConnections [private]

The lbl xmpp connections.

Definition at line 60 of file XmppPanel.java.

Referenced by XmppPanel().

4.5.4.5 HashMap<String, String> XmppPanel.m_accounts = new HashMap<>() [private]

The m_accounts.

Definition at line 39 of file XmppPanel.java.

Referenced by connect(), and XmppPanel().

4.5.4.6 ConnectionConfiguration XmppPanel.m_config [private]

The m_config.

Definition at line 72 of file XmppPanel.java.

Referenced by connect(), and XmppPanel().

4.5.4.7 JList XmppPanel.m_connectionList [private]

The m_connection list.

Definition at line 48 of file XmppPanel.java.

Referenced by XmppPanel().

4.5.4.8 HashMap<String, WTFConnection> XmppPanel.m_connections = new HashMap<>() [private]

The m_connections.

Definition at line 36 of file XmppPanel.java.

Referenced by connect(), disconnect(), and handleSIPMessage().

4.5.4.9 DefaultListModel<WTFConnection> XmppPanel.m_listModel [private]

The m_list model.

Definition at line 57 of file XmppPanel.java.

Referenced by connect(), and XmppPanel().

4.5.4.10 JTextArea XmppPanel.m_logArea [private]

The m_log area.

Definition at line 51 of file XmppPanel.java.

Referenced by handleSIPMessage(), printMessage(), and XmppPanel().

4.5.4.11 HashMap<String, String> XmppPanel.m_relation_in = new HashMap<>() [private]

The m_relation_in.

Definition at line 42 of file XmppPanel.java.

Referenced by forwardMessage(), printMessage(), and XmppPanel().

4.5.4.12 HashMap<String, String> XmppPanel.m_relation_out = new HashMap<>() [private]

The m_relation_out.

Definition at line 45 of file XmppPanel.java.

Referenced by handleSIPMessage(), and XmppPanel().

4.5.4.13 JScrollPane XmppPanel.scrollPane_2 [private]

The scroll pane_2.

Definition at line 54 of file XmppPanel.java.

Referenced by XmppPanel().

4.5.4.14 final long XmppPanel.serialVersionUID = 5657339239939443076L [static], [private]

The Constant serialVersionUID.

Definition at line 33 of file XmppPanel.java.

4.5.4.15 ConnectionConfiguration XmppPanel.xmppConfig [private]

The xmpp config.

Definition at line 69 of file XmppPanel.java.

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

- src/XmppPanel.java

Chapter 5

File Documentation

5.1 src/Server.java File Reference

Classes

- class Server

5.2 src/WTFConnection.java File Reference

Classes

- class WTFConnection

5.3 src/XmppConnectionHandler.java File Reference

Classes

- class XmppConnectionHandler

5.4 src/XmppGateway.java File Reference

Classes

- class XmppGateway

5.5 src/XmppPanel.java File Reference

Classes

- class XmppPanel

Index