libskaro Documentation

Release 0.2

Alexandr Danilov

May 15, 2016

CHAPTER 1

Quick Links

- libskaro on GitLab
- SkarIM-QML on GitLab
- SkarIM-Server on BitBucket

CHAPTER 2

Documentation Contents

2.1 Installation

libskaro requires Python >= 3.5 and is known to work on Linux, Mac OS X, and Windows (with Cygwin).

2.1.1 Python Package Index (PyPI)

libskaro is listed in PyPI, and may be installed using pip:

```
pip3 install libskaro
```

2.1.2 Install from Source

The libskaro code is available from GitLab. Either download and extract a libskaro release archive, or clone the libskaro repository:

```
git clone https://gitlab.com/exarh-team/libskaro.git
```

Switch to the hangups directory and install the package:

```
cd hangups
python3 setup.py install
```

2.2 API

2.2.1 Public methods

Abstract class for writing chat clients.

class `libskaro.client.`**`Client`**(*url*)

Instant messaging client for skaro.

Maintains a connections to the server, emits events, and accepts commands.

`__init__`(*url*)

Create new client.

Parameters **`url`** (*`string`*) – the url address of the server.

Registration / Authorization

connect()
This function is called to connect to the server and runs all the time, until the connection to the server is actively.

authorize(*data*)
The authorization request of the user.

The client uses the stored token (see. _check_connection) and hashes it user password as the key for the HMAC-SHA512 hashing.

The resulting hash it sends as a user password in the message to the server type authorize

Parameters **data** (*dict*) – {'login': __YOUR_USERNAME__, 'password': __YOUR_PASSWORD__}

register_type()
Requests a registration type on the server: open, by invitation, closed.

session_restore(*sid*)
Tries to restore the broken session. In the result of the authorization (event 'on_received_user_data') we received the session identifier (SID), which is used in this query. The session can be restored, if the gap was no more than 5 minutes.

Parameters **sid** (*string*) – the session identifier (SID).

Work with contacts

get_contacts()
Requests a list and information about all the users whom we know.

Work with messages

get_history(*login*)
Requests a message history with another user.

send_message(*login*, *message*)
Produces the sending of a message to another user.

Parameters

- **login** (*string*) – the login of the user for whom the message is intended.
- **message** (*string*) – a message that will be sent.

2.2.2 Private methods

Registration / Authorization

Client.**_check_connection**(*data*)
It called immediately after the connection and contains a token used for authentication.

Parameters **data** (*dict*) – received information.

Client.**_authorize_on_request**(*data*)

The success of the operation defined by the field 'status' and depends on whether coincided: the hash sent by the client, and the hash generated by the server on the same algorithm as that of the client.

Thus hashes coincide only with the same hash key - the user's password, but the password over the communication channel is not transmitted.

Parameters **data** (*dict*) – received information.

Client.**_register_type_on_request**(*data*)

The answer to the query 'register_type'

Parameters **data** (*dict*) – received information.

Client.**_session_restore_on_request**(*data*)

If successful, it performs the initialization procedure of the user.

Parameters **data** (*dict*) – received information.

Work with contacts

Client.**_get_contacts_on_request**(*data*)

The answer to the query 'register_type'

Parameters **data** (*dict*) – received information.

Work with messages

Client.**_get_history_on_request**(*data*)

The answer to the query 'register_type'

Parameters **data** (*dict*) – received information.

Client.**_send_message_on_request**(*data*)

The answer to the query 'register_type'

Parameters **data** (*dict*) – received information.

Client.**_show_message**(*data*)

It called when someone sends you a new message.

Parameters **data** (*dict*) – received information.

Other

Client.**_info**(*data*, *on_request=None*)

The server wants to communicate some information.

Parameters

- **data** (*dict*) – received information.
- **on_request** (*string*) – the type of query that reports information.

Client.**_error**(*data*)

The server reports the error occurred.

Parameters **data** (*dict*) – received information.

`Client.`**`_received_message`**(*data*)

Called if there was a new response from the server. Depending on the type of response determines what to do with it.

Parameters **`data`** (*`bytearray`*) – received information.

`Client.`**`_init_user`**(*data*)

Initializes user (operation performs the primary input into account).

Parameters **`data`** (*`dict`*) – received information.

`Client.`**`__write_client__`**(*data*)

Sending a data to the server.

Parameters **`data`** (*`dict`*) – transmitted information.

CHAPTER 3

Indices and tables

- genindex
- modindex
- search

Python Module Index

Index