ATutor Chat Redesign

XMPP Chat 2.0 Developer guide

Done by: Casian Olga
Mentored by: Harris Wong

Idea description

Before the ATutor chat worked in synchronous mode that was not the optimal solution in terms of both: user experience and efficiency. However, the technology evolved since the time AChat was developed. The goal of this project was to make a new version of the chat based on XMPP protocol and WAI-ARIA live regions introducing more efficient data transfer, larger feature set, accessible and intuitive interface.

Currently the XMPP Chat has the following features:

- One-to-one messaging and Multi User Chat (MUC) among course members;
- Roster management (tracking participants' presence);
- Subscription management (automatically adding new joined users);
- Saving user's nickname and password after first authorization;
- Saving history for both private and group chat messages;
- Highly secured authorization on third-party XMPP server;
- Offline messaging support even for group chats;
- Message encryption (on ATutor server);
- WAI-ARIA accessible regions;
- User friendly interface ("started typing" detection, ajax content loading, links highlighting, etc.).

Requirements

The XMPP Chat uses the tools that provide all modern browsers: JavaScript and image support should be turned on. The PHP requirements are the same as for ATutor, 5.0.2+. This version is compatible with ATutor 2.1.1 and newer.

Tested environment

Currently the XMPP Chat was successfully tested on:

- Linux Mint 12 (Ubuntu)
    - Browsers: Firefox 14.0.1, Chromium 18.0.1025.168, Opera 12.01;
    - Screen readers: Orca;
- Windows 7
    - Browsers: Firefox 12.0.1, Google Chrome 21.0.1180.75 m, Opera 12.01, Internet Explorer 9;
    - Screen readers: NVDA, JAWS.

General notes

The XMPP Chat uses modified, extended and fixed code snippets from the book "Professional XMPP Programming with JavaScript and jQuery" by Jack Moffitt, in particular examples from the second part, chapters 4: "Exploring the XMPP protocol: a debugging console", 6: "Talking to friends: one-on-one chat" and 8: "Group chatting: a multi user chat client".
The code uses XEP-0045: Multi-User Chat [http://xmpp.org/extensions/xep-0045.html] and XEP-0085: Chat State Notifications [http://xmpp.org/extensions/xep-0085.html] XMPP protocol extensions.
Please see the comments in the code for more information on the implementation.

Deliverables

- Root module folder: contains the files needed for module installation and uninstallation, documentation, index page of the module and CSS style sheets;
- ***ajax*** module folder: contains PHP files and libraries used in ajax requests;
- ***includes*** module folder: contains tab content of the module and side box related files;
- ***js*** module folder: contains JavaScript classes and libraries that handle XMPP protocol events and reflect them on the interface; *xmpp_consile.js* has console that displays all the XMPP traffic, it can be activated by uncommenting *#peek* div in *chat_new/index.php* file.

Used libraries

- ***jQuery***: included earlier in ATutor;
- ***jQuery UI***: provides tabs and dialogs used in the project; [https://ajax.googleapis.com/ajax/libs/jqueryui/1.8.18/jquery-ui.min.js];
- ***Strophe***: JavaScript implementation of XMPP protocol; MIT license, [http://strophe.im/strophejs/];
- ***flXHR***: a Flash replacement for the standard XMLHttpRequest API, normally, a JavaScript application cannot talk to external servers, but with the help of Flash and flXHR, Strophe can overcome this restriction; Public Domain license, [https://github.com/flensed/flXHR];
- ***SCRAM-SHA-1 authentication support for Strophe***: currently there is no released version of Strophe that supports highest authentication mechanism XMPP provides, however this code adds the implementation; [https://github.com/metajack/strophejs/pull/43];
- ***moment.js***: a JavScript date library for parsing, manipulating, and formatting dates; is used for all dates in the module; MIT license, [http://momentjs.com/];
- ***mcrypto***: a PHP script (for PHP 5.0+) for encrypting and decrypting data that uses Mcrypt library; Public Domain license, [http://codomaza.com/script/skript-shifrovanija-dannykh-s-pomoshhju-biblioteki-mcript].

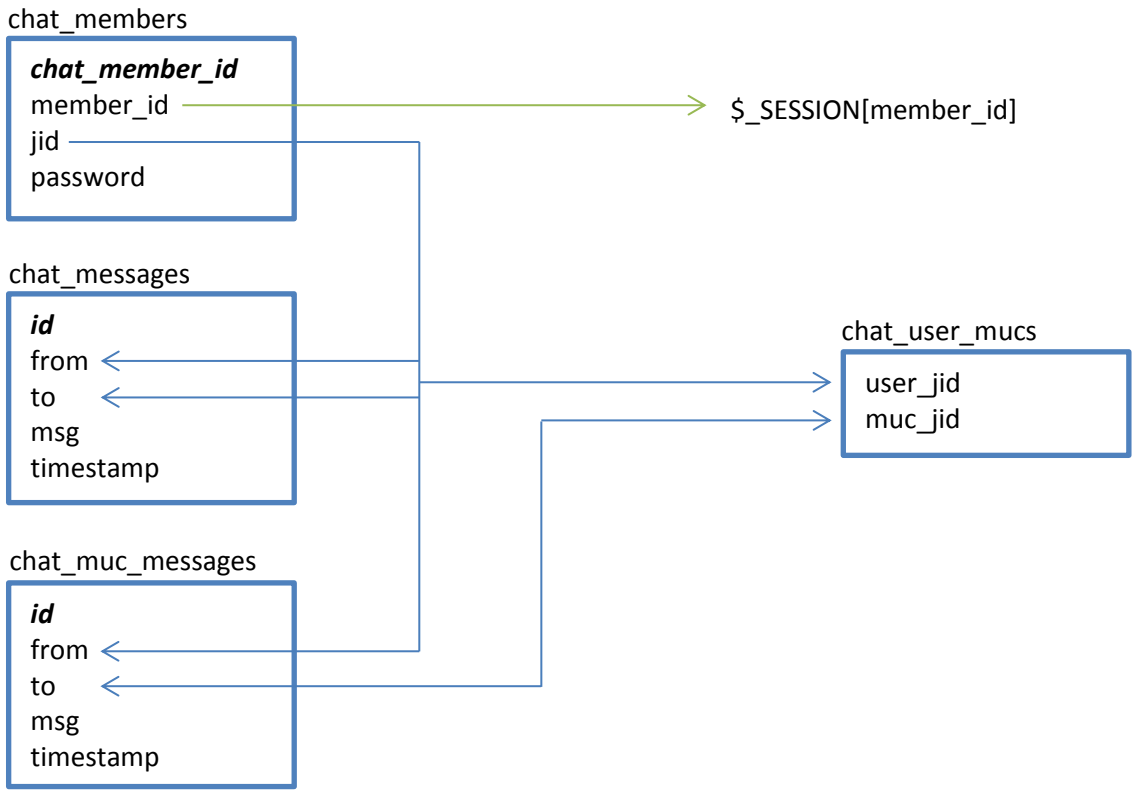
Database tables

During the module installation four tables are created in the ATutor database, all of them have prefix *chat*:

- ***chat_members***: contains account data for connection to XMPP server;
- ***chat_messages***: contains all private messages;
- ***chat_muc_messages***: contains all group chat messages;

– ***chat_user_mucs***: stores group chat IDs for each user where he/she is a member.

For more details see the following database diagram:



Troubleshooting

The delivered version of the chat uses two third-party servers: an open-source XMPP server, *talkr.im* [https://www.talkr.im/], and a BOSH connection manager [http://bosh.metajack.im:5280/xmpp-httpbind] that was provided by the author of the book "Professional XMPP Programming with JavaScript and jQuery", Jack Moffitt only for the development phases. It is highly recommended to set at least one connection manager for the chat on ATutor server.

For this I would recommend the *punjab* [https://github.com/twonds/punjab], a standalone BOSH connection manager. The following procedure describes the usage and installation on Linux environment (more detailed description can be found in the book by Jack Moffitt, Appendix B):

1. Get the latest available version of the *punjab* from the github page.
2. Install the dependencies and follow the steps described in INSTALL.txt [https://github.com/twonds/punjab/blob/master/INSTALL.txt].
3. The only thing that must be edited in the module code is the URL in *chat_new/js/xmpp_client.js* in the beginning of *'connect'* event. By default the new URL should look similar to *http://localhost:5280/xmpp-httpbind* (replace localhost with the domain name).