Chat Box 6:

# Features:

1. Multi-room support, via URL or Chat command
2. Configurable announcements display period, room buffer to limit server used resources, allowed time gap between two messages, Time to go AFK (*Away From Keyboard*), Time to go Offline, Emotions directory, Update interval, AFK update interval, Max emotions to show (*extra ones are accessible via a special button*), Sign in page, Kick out page, Avatar URL template, Sounds notifications
3. Greetings messages
4. Announcements
5. Guest mode (*visitors can read public messages but not allowed to send*)
6. Ban users
7. Kick out users
8. Censor custom words
9. Changing user’s display name (Alias name)
10. Toggle on / off some feature such as (*Gust mode, Censored words, Greetings, Announcements, Emotions*)
11. Customized groups and permissions (*allowed commands and group power*)
12. Emotions panel
13. Private messaging
14. Online users panel
15. Administration commands such as (*clean, clear, ban, unban, banlist, kick, unkick, kicklist, op, deop, users, groups, rooms, cr, alias, help*)
16. Command syntax error hint
17. Configurable command prefix
18. Sound notification for received message (*public , private*) and for online users status change
19. Configurable message queuing time, adds enough time between two messages
20. Configurable connection-error retries
21. Configurable request timeout period
22. Flat-text based, no Database is needed
23. Error Logging, user will be notified in case of any error

# Configurations:

**1- Adjust sa user (*super admin*):**open the file **data\groupusers.txt**

sa super super 2012-03-26 00:00 am

Adjust its contents (*tab separated values*)

Above values refer to:

(**sa**) the group name we have in the **$PERMISSIONS** array (*at the top with all permissions*)

(**super**) the username assigned to the **sa** group

(**super**) the username who assigned this user to this group

(**2012-03-26 00:00 am**) the date the user is assigned to this group

This user will have the full control over the chat and all other users can be added from the chat app itself  
currently there are 4 predefined groups (*sa, admin, mod and member which is the default group for any logged in user if he hasn’t assigned to any group*)

**2- Configure the application:**open the **config.php** file  
the variables in there are mostly self-explanatory

**3- Adding more rooms:**

open the **config.php** file

under the Available rooms section you can add room to the **$ROOMS** array (*room code -> room display name*)

for each room you add you have to add two files to the **data** folder in this convention:

room\_*x*.*ext*, onlineusers\_*x*.*ext* where **x** is the room code and **ext** is the extension you specify in the **config.php** file for **ROOM\_FILE** and **ONLINE\_FILE**

if the room code is empty (‘’) there is no need for the underscore ( \_ ) at the end of these file names

you can use the chat commands (rooms, cr) to change the current room, or by sending the room code as a part of the query string using (**rm**) as variable name, something like:

localhost/chat6/?rm=*x*

where x is the room code you specify in **$ROOMS** array

**4- Modify the group permissions:**at the bottom of the **config.php** file you can update the **$PERMISSIONS** array as needed  
you can add groups as needed but make sure to keep the **sa** at the top and the **DEFAULT\_GROUP** at the bottom of the **$PERMISSIONS**array

**sa**group (*or whatever you want to call it*) must have all permissions, and the must have the minimum amount of permissions

All groups but **sa** must have less permissions than **sa**

The (ban, unban, kick, unkick, op, deop) commands depend on the group power to be executed

The order of the group in the **$PERMISSIONS** array gives it the power (*first one has the most power*)

**5- Configure the client:**open the **client\_config.php** file  
you can add the emotions, change the message update interval, messages queue time, max emotions to display (*a new icon will be added next to them which allows you to browse hidden ones if any),* sign in page, sign in message, kick out page, the avatar template (*currently set to match the FB template*)

**6- Announcements:**open **data/announcements.txt** file  
add your announcements (*1 per line*)  
if you want to make multi-row announcement, use the**<br />** html tag  
you can design you announcement as you usually do and once it's ok remove all line breaks from it using notepad++ or any other capable editor to convert it to a one line entry  
announcement time can be adjusted in the **config.php** file

**7- Greetings:**open **data/greetings.txt**file  
add your greeting (*1 per line*)  
you can use user info (**{username}, {user\_display}, {group}** ) in the greeting message

these templates will be replaced with their corresponding values once loaded

**8- Censored words:**open the **censoredwords.txt** file  
add one word per line to block them from appearing to users

**9- Change group CSS style:**open the **chat.css** file  
there are two places to change the group styles (*for the online users and messages*)  
whatever are the groups you already have, you may want to add a style for each one of them  
the style rule convention is: **grp\_xxx**where **xxx** is the group name you declared inside the **$PERMISSIONS** array  
another predefined group is the bot group (**grp\_bot**), you may want to change its style as well  
search for something like **.text .grp\_bot** (*to style the group in the message section*) and you'll see all other group rules (*use the same concept for other groups if you want to add any new ones*)  
search for something lime**#usersOnLine .grp\_member** to style the group in the online users section   
search for **#usersOnLine .away** and **#usersOnLine .online** to style the away from keyboard / online users  
search for **.text .greeting** and **.text .announcement** to style the greetings and announcements  
search for **.text.private** and **.text .note** to style private messages and the bot notes

# Integration:

This must be done after configuring necessary variables in the **config.php** and **client\_config.php** files.

1- Adjust the file paths, chat settings, features and session variable names as needed

2 - Point the **SIGNIN\_PAGE** in the **client\_config.php** to your application sign in page (*or you can use included* ***fb\_signin.php***)

To make use of the included Facebook login script (**fb\_signin.php)** use the following code:

"javascript:openWindow('fb\_signin.php','','width=410,height=300')"

3 - After a successful login, your sign in page is required to save at least the (**USERNAME**, **USER\_AUTH**) to a session, and optionally you can save the (**USER\_ID**, **USER\_DISPLAY**) too (*session names defined in the* ***config.php*** *file*)

Use **authAlgo()** function to generate **USER\_AUTH** code, *currently this algorithm is generating MD5 (username)*

4 – In case you want to modify or add some functionality, use the defined session names (**USER\_ID, USERNAME** ...) instead of their string values to make it easier for you to adjust them all from one place (**config.php**)

5- You can use your own method to generate the authentication key by updating the **authAlgo()** function inside the **core.php** file

6- Use the index.htm file as a template if you want to include the chat box directly in your webpage (*check file paths in the index.htm and configuration files they must point to the correct locations*), or use an **iframe** to host the **index.html**

7- You can adjust **INIT\_SCRIPT** in the config.php file to point to specify a script to be executed before initiating the chat box which will allow us to perform some tasks like retrieving user's login info using some APIs (*i.e Facebook API*) in order to implement an auto login functionality (***fb\_init.php*** *will implement auto login using Facebook API*)

**Notes:**

If you want to login using Facebook account you have to first create a Facebook application and point it to your chat box URL and then adjust the **appId** and **secret** codes in both **fb\_signin.php** and **fb\_init.php** to the ones provided you by Facebook

# Permissions and Group power:

Permissions are simply the command set assigned to a specific group

Group power is what prevent a user from (*ban, unban, kick, unkick, op, deop*) another user in the same or more powerful group even if he has the required permission to do that

So, assuming we have two groups (**A***,* **B**) with ban command in their permissions, and group (**A**) comes before group (**B**) in the **$PERMISSIONS** array

A user from group (**A**) can ban a user from group (**B**) but not the vice-versa

# Files and Folders structure:

|  |  |
| --- | --- |
| **Directory** | **Contents** |
| chat6 | index.htm  kicked.htm  chat.css  config.php  client\_config.php  server.php  core.php  commands.php  soundPlayer.swf  fb\_signin.php  fb\_init.php  signin.php (fake) |
| chat6/data | greetings.txt  announcements.txt  censoredwords.txt  bannedusers.txt  kickedusers.txt  groupusers.txt  onlineusers.txt (\*)  room.txt (\*)  alias.txt |
| chat6/js | ajax.js  client.js |
| chat6/sounds | new\_message.mp3  private\_message.mp3  users\_status.mp3 |
| chat6/fb\_sdk | base\_facebook.php  facebook.php  fb\_ca\_chain\_bundle.crt |
| chat6/emotions | Smiley or similar |
| chat6/images | Styling related images |

( **\*** ) you may have multiple file with similar naming in case of using multiple rooms (*see Configuration / Adding more rooms section*)

# Client’s Functions:

|  |  |  |
| --- | --- | --- |
| **Function** | **Arguments** | **Description** |
| trim(**str**) | str: String | Remove suffixed and prefixed white spaces |
| getUserAvatar(**user**) | user: JS object, contains user info | Replace user info in the configured avatar template |
| openWindow(**url**,**name**,**specs**) | url: String, url to the page  name: String, the name of the page  specs: String, window specs | Open popup window, can be used in the sign in page URL |
| dom(**id**) | id: String, the ID of an HTML tag | Return HTML object by its ID |
| queryString(**p**) | p: String, parameter name you want to read from the url | Return parameter value, or null if not found |
| insertAtCursor(**field**, **value**) | field: HTML object, the message text  value: String, value to be inserted | Insert emotion code at cursor position |
| formatMessage(**m**) | m: String | Replace smiley codes and [url] tags with their HTML equivalent codes |
| loadEmotions() |  | Load configured emotions to the chat box interface and assign necessary click events |
| ajax\_error(**err, isTimeout**) | err: String / Object, error data  isTimeout: Boolean, | Handle server errors / request retries |
| ajax\_timeout() |  | Handle request timeout |
| reconnect() |  | Reconnect to the server by sending the data where the request failed |
| callServer(**data**) | data: JS object, contains name-value pairs to be sent to server | Handle all server requests |
| ResponseHandler(**res**) | res: JS object, contains JSON received from server | Handle all server responses |
| processCommand(**sc**) | sc: Array, contains server command and probably some other arguments | Handle commands sent by server to client |
| processMessages(**nm**) | nm: Array, new messages | Display messages loaded from server |
| processOnlineUsers(**ol**) | ol: Array, online users | Display online users and assign necessary click events |
| initiateChatBox() |  | Load emotions, client configurations, guest mode and user information |
| signin(**user**) | user: JS object, contains user info | Sign in specified user (must be already authenticated) |
| signout() |  | Sign out current user |
| sendMessage(**q**) | q: Boolean, the message is sent from the queuing list | Send message to specified users or issue a command |
| check() |  | Check for new messages or user status changes |
| clearToUser() |  | Clear ToUser field (send to all) |
| getPlayer(**id**) | id: String, the ID of an HTML tag | Get SWF player object |
| sp\_ready() |  | SWF player is ready (called from within the player once loaded) |
| sp\_play(**name**) | name: String, notification name | Play a notification sound by its name |

# Server’s Functions:

|  |  |  |
| --- | --- | --- |
| **Function** | **Arguments** | **Description** |
| error\_function($level,$msg,$file,$line) | $level: Integer, error level  $msg: String, error message  $file: String, file path where error occurred  $line: Integer, line number | Handle errors may occur (i.e. file paths or permissions) |
| ltime($format="Y-m-d H i", $time=0) | $format: String, date format  $time: Integer, timestamp | Convert UTC local time |
| authenticated($user) | $user: String, username | Check user authentication key(must be already saved to a session) |
| censoredWords($str) | $str: String, message to check | Replace censored words with first and last letters and stars in between |
| getAlias($user, $default=null) | $user: String username  $default: String, default display name | Return user defined display name |
| getGroup($username=null) | $username: String, username | Return user group / group power |
| groupPower($username) | $username: String, username | Return group power |
| postMessage($id,$from,$fromDisp,$group,$to,$msg) | $id: Integer, user ID  $from: String, username  $fromDisp: String, user display name  $group: String, user group  $to: String, targeted users (comma separated)  $msg: String, the message | Save the message to the room or directly to the response array |
| getNewMessages($room=null) | $room: Array, room file array | Load new messages to the response array |
| updateUser() |  | Update user last post time (or delete a user) |
| getOnlineUsers($onlineusers=null) | $onlineusers: Array, online users array | Load online users list to server response and return the new message indicator |
| isBanned() |  | Load the banning message to server response |
| isKicked() |  | Return the kicking message or false otherwise |
| getGreeting() |  | Return random greeting message |
| getAnnouncements() |  | Load the announcement to server response and update announcement counter and time |
| doCommand($com,$msg) | $com: String, command  $msg: String, the message | Execute command considering group allowed permission and group power and load execution result to server response |

# Client to Server Commands:

|  |  |  |
| --- | --- | --- |
| **Command** | **Parameters** | **Description** |
| init | o: String, ‘init’  r: Boolean optional, is reloading | Load emotions, client configurations, check guest mode, user info |
| signin | o: String, ‘signin’  id: Int optional, user ID  u: String, username  ud: String optional, user display name  z: Decimal auto, timezone  a: String, authentication key | Check user authentication, save user info |
| signout | o: String, ‘signout’ | Clear user info |
| send | o: String, ‘send’  m: String, message  to: String optional, to users (comma separated) | Send a message to specified user (or to all) |
| check | o: String, ’check’ | Check for new messages or users status changes |

# Server to Client Commands:

|  |  |  |
| --- | --- | --- |
| **Command** | **Parameters** | **Description** |
| init | EMOTIONS: Array, emotion codes and images  CONFIG: Array, client configurations  GUEST\_MOD: Boolean, guest mode enabled/disabled  (User Info): Array, currently saved user info | Load emotions, client configurations, check guest mode, user info, happens on page load |
| signin |  | Redirect the user to sign in page if the authentication key is invalid |
| logged | (User Info): Array, currently saved user info | Sign in is successful and the user is logged in, user can send and receive messages from this point on |
| reload | GUEST\_MOD: Boolean, guest mode enabled/disabled | Reinitiate the application by calling ‘init’ command on the client side, happens after signing out |
| sleep |  | Stop the client from checking the server for new messages, happens when a user reaches Offline period |
| error | (Error Message) | Show error message, happens when unhandled error occurs |
| wakeup |  | Wake up the client from sleep mode and resume checking the server for new messages |
| away |  | Change the server checking interval when a user reaches the AFK period |
| kicked | (Reason) | Redirect the user to the kick out page |
| Clean |  | Clean client’s chat box screen |
| room | ROOM\_CODE: String, room code  ROOM\_NAME: String, room display name | Change the room info after changing the room |
| displayname | ud: String, user’s new alias name | Update user display name after calling the alias chat command |

# High Level Flowcharts:











