

# User profiles

## Problem

Convenience of having different communication contexts in a single client without shared metadata.

Currently requires changing chat database and restarting chat.

Events (notifications) for databases other than current are not received.

## Solution

Support for multiple user profiles in a single database.

Separate transport connections with the same servers not shared between profiles.

## Design

### API

- new users are created using existing CreateActiveUser API
- APIListUsers
- APISetActiveUser UserId
- CRUsersList {users :: [User]}
- terminal APIs

### Subscriptions

- get connections for all users, subscribe to all - to receive events for users other than active
- map subscriptions to users?
- option to subscribe to connections for only one user? StartChat {allUsers :: Bool} API
- if more than one user profiles - start with last active user (persist?), or always load list of users first and wait for choice? same for terminal?

### Chat items expiration

- store supports this configuration per user

- expire chat items for all users with different ttl
  - in runExpireCIs get list of users, for each - getChatItemTTL and run expireChatItems

## Disappearing messages

- cleanupManager - remove parameterization by User
- either getTimedItems and start deletion threads for all users, or similar to expiration get list of users and run cleanupTimedItems per user

## Events

- events need to have information for which user they happened

```
haskell data UserChatResponse = UserChatResponse { user :: User, chatResponse :: ChatResponse }
```

- replace ChatResponse with UserChatResponse in APIResponse, toView, etc.
- in agentSubscriber don't get currentUser from controller, instead either:
  - get user from database by agent connection id
  - get from subscriptions user-connections map - either requires scan or keys and values have to be inverted, also key would have to be agent connection id not database connection id - probably easier/better to read from db
- non active user profile can be shown in notifications
- interactions via notifications - prohibit for simplicity? or change APIs to allow specifying User?
- changes to chat model are applied only for current user

## ChatController

- currentCalls:
  - change currentCalls key? currentCalls :: TMap (UserId, ContactId) Call
  - restoreCalls - for all users?
  - interaction via notification - prohibit for non active user? change current user before accepting?
- incognitoMode - save setting or share between users?

- when changing active user reset activeTo?
- in newChatController when creating smpAgent - getSMPServers have to get servers for all users, or depending on allUsers flag in StartChat; // drop known\_servers table?
- AgentConfig should depend on network configuration per user?
- double check other state

## Storage

- schema already has support for multiple users
- persist last active user?
- persist user settings? (see below)

## Frontend

- view with list of user profiles available from settings, change active user from there
- do we need additional information in that view? for example, number of unread messages
- network configuration is stored in app preferences - if unchanged will be the same across users
  - persist in database per user and load?
  - different configuration across platforms
  - save network settings in separate preference for each user as json? dynamic preferences or predefined number of users?
- same for other settings - auto-accept images, send link previews, etc.

## Terminal view

- for users other than active view responses to have indication that it's for a different user profile, e.g. [<user\_name>], [user: <user\_name>]
- don't set activeTo for non active user profiles
- only way to reply to a message in other user profile is to manually switch current active user first?