

User System

Status of this Memo

This memo provides information for the Xaraya community. It does not specify an Xaraya standard of any kind. Distribution of this memo is unlimited.

Copyright Notice

Copyright © The Digital Development Foundation (2002). All Rights Reserved.

Abstract

The contents of this RFC contain the literal content of the old plain text version of RFC-0016

When time is a less scarcer good, someone might convert the plain text into structured XML so we can benefit from it.

Table of Contents

| | |
|---|----------|
| 1 Requirements for the new Users System..... | 3 |
| 2 Solution Proposals - Functions..... | 5 |

1. Requirements for the new Users System

I've gone through the SF feature requests and compiled a list of requirements for the new User module and a flow for the new user registration process. They're in pretty rough form but should be understandable, please take a look through and let me know if I've missed anything.

New user flow

- agree to terms and conditions (if required)
- get account details
- create user account (account inactive)
- send user confirmation email (if required)
- receive reply from confirmation email (if required)
- set user account to pending
- request activation from admin (if required)
- receive activation from admin (if required)
- activate account

Admin controls (each can be on or off, set by admin for all users)

- get user to agree to terms and conditions before allowing them to proceed with creating an account
- allow user to choose their own password when registering
- get user to respond to a registration confirmation email before activating their account
- explicit admin approval of account registration required for activation
- allow user to set themselves to be invisible in members list
- send welcome email on registration
- enable/disable user points system

Admin actions

- modify avatar choice (avatars might be in a separate module, haven't decided yet)
- modify registration confirmation email (if set)
- modify terms and conditions (if set)
- modify welcome email (if set)
- activate/deactivate a specific user account (inactive = account still present but user cannot log on with it)
- expire user (== inactive but separate state needed for time-based subscriptions)
- delete user
- edit a user's details
- modify list of disallowed email domains during registration
- modify list of disallowed usernames during registration
- change status of a user field (required/optional/hidden)
- create/modify/delete a user field
- change expiry date for user
- reset a user's password
- allow expiry date for users

- allow user to collect points for specific actions (e.g. posting a news item, answering an FAQ, posting a comment, ...)
- change number of points generated for each action
- change names/numbers for points rankings
- email some/all users
- approve a user

Admin views

- users in order of registration date
- users waiting approval (if set)
- users in order of last login

User actions

- create an account
- login
- set themselves to be visible/invisible in members list
- edit account details (including themes) (and module settings?)
- change password
- retrieve lost password (question/answer?)

User views

- visible users

Blocks

- login
- who's online(?)
- our latest users

Hooks

- user activation
- user logon

User states like this define whether a user is capable of logging on or not. So we might have states like:

- disabled - admin deactivated
- expired - time deactivated
- pending - awaiting admin approval
- unconfirmed - awaiting confirmation that email address of user is correct
- active - normal state

And the user could only log on if the account was active. This would allow subscription-based sites along with better admin control than at current, where the only way an admin could lock out a user is to delete their account or change their password, neither of which would give a very useful error message to the user.

Group membership would probably come in to play for viewing content on the site, but the actual state of the user is a separate thing.

2. Solution Proposals - Functions

Modules should be able to plug into the users module to offer configuration screens. This could be accomplished with a hook that is registered by the module for the display of the user settings screen in the users module. The search module uses a similar implementation.