Proposal OpenID Support

Status	Proposal under development
Special Note	Google Summer of Code 2008 project
Target Release	4.1
JIRA Issue	ROL-1733
Original Authors	Tatyana Tokareva, Dave Johnson

Abstract

OpenID is a technology that allows users to use one username/password pair to login to large number of web sites, but to never have to reveal their password to any of those sites. Instead of establishing a username/password pair on each site, a user instead registers with an **OpenID provider** site (e.g. claimid.com) and receives in return a unique **OpenID identifier** URL. When logging into a site that supports OpenID logins, known as an **OpenID consumer** site, the user specifies this URL to the consumer site, is directed to the provider site to login and then back to the consumer site in a logged in state.

This proposal is for adding OpenID support to Roller so that Roller can act as an OpenID consumer.

Requirements

Here are the basic requirements to be satisfied by this proposal:

- Allow new users to register and login via OpenID
- Allow existing users to login via OpenID
- Allow site operators to choose one of three modes of operation:
    - Disabled: no OpenID support and no evidence of OpenID in the Roller web UI
    - Hybrid: allow users to login via either username/password or OpenID identifier
    - Only: require users to login only via OpenID identifier

Issues

Here are the major issues we discussed during the development of this proposal:

- Allow users to specify more than one OpenID? Decision: no, one is sufficient
- Allow commenting users to login via OpenID? Decision: good idea, but we'll do this later
- Add OpenID specific methods to Roller's User Manager? Decision: no, use generic user attributes to store each user's OpenID identifier instead.

Design

Here is an overview of the design of the OpenID consumer support in Roller:

Use Spring Security's OpenID support

We will use Spring Security v2.0.3 (the latest release) and its OpenID support feature.

New database table and POJO

We will add a new database table 'roller_userattribute' and a new POJO UserAttribute to store each user's OpenID identifier URL. Later, this table can be used to associated other data with each user.

New UserManager methods

Add methods for setting and getting user attributes.

New startup properties

Add a new startup property called 'authentication.openid' with three possible values: disabled, hybrid and only. See the requirements for the meanings of these strings.

Changes to registration form

Add an OpenID identifier field that will be shown in hybrid and only modes. Here's what the form will look like in hybrid mode:

Changes to login form

Add an OpenID identifier field that will be shown in hybrid and only modes. Here's what the form will look like in hybrid mode:

Changes to user profile form

Add an OpenID identifier field that will be shown in hybrid and only modes.

Project Plan

1. Examine the application structure and the features of the existing authorization system, built using the Spring Acegi framework. Understand, what types of security filters are used and how they can be replaced. Outline, what features to change and what to edit in the existing system.
2. Plan the main functionality of the project and install required libraries to the system.
3. Release the main functionality - add necessary classes, scripts, database tables and Spring Security filters.
4. Change the visual interface of the application (add authentication prompt to ask users for their OpenID identifier rather than their username and password).
5. Test the system and write necessary documentation.

Timeline

- 26th of May
  - Initial planning of the project and research
  - Deliverable: Document of detailed description of the project.

- August, 11th
  - To this point the code of the application will be given.
  - Deliverable: Coding necessary to implement the OpenID authentication

- August, 18th (deadline)
  - In this point, corrections would be done that make lack to the previous code.
  - Deliverable: Documentation on the project.

Comments

Please comment on the Roller-dev mailing list.