

Wirehose-Access Control

WireHose defines a delegate interface which can use to customize how users are authenticated, how guest users get created, and whether or not users can view, edit or delete objects. The WireHoseEngageSupport framework provides an implementation which uses WHTag subclasses to model roles-based access control, with users, groups, operations and inheritable permissions.

Authentication

[WHApplicationHelper](#) will automatically authenticate users based on username and password or unique login cookies or request headers. If a user can't be found, a guest user will be used. You can customize this behavior by implementing delegate methods which allow you to inspect whatever form values, cookies or headers you find in an incoming request to determine the user for a session.

Access control

The [WHApplicationHelper.Delegate](#) interface also defines methods such as `userCanViewObject` and `userCanEditObject`, which you can implement to control what users are allowed to do.

The [com.wirehose.base.engage](#) package uses tags to implement roles-based access control, with users, groups, operations and inheritable permissions for taggable objects. [WHEngageTag](#) is a WHTag subclass which is taggable and indexable, so you can control access to tags. [WHEngageUser](#) represents users which can belong to multiple groups. [WHGroup](#) is a WHTag subclass which represents groups that can be arranged in an arbitrary hierarchy.

Permissions, represented by the [WHPermission](#) class, are tags which are assigned to taggable objects to indicate an association between a group and a [WHOperation](#) object. Permissions can be marked as inheritable, which means you can assign a permission to a tag, and any resources tagged with that tag will inherit the assigned permission.

Starting content used with permission of Gary Teter. WireHose and the eyeball-and-arrows logo are trademarks of Gary Teter.