

Table of Contents

Foswiki::AccessControlException.....	1
package Foswiki::AccessControlException.....	2
ClassMethod new(\$mode, \$user, \$web, \$topic, \$reason).....	3
ObjectMethod stringify() -> \$string.....	3

Foswiki::AccessControlException

package Foswiki::AccessControlException

Exception used raise an access control violation. This exception has the following fields:

- `web` - the web which was being accessed
- `topic` - the topic being accessed (if any)
- `user` - canonical username of the person doing the accessing. Use the methods of the `Foswiki::Users` class to get more information about the user.
- `mode` - the access mode e.g. `CHANGE`, `VIEW` etc
- `reason` a text string giving the reason for the refusal.

The exception may be thrown by plugins. If a plugin throws the exception, it will normally be caught and the browser redirected to a login screen (if the user is not logged in) or reported (if they are and just don't have access).

API version \$Date: 2009-05-16 12:36:23 +0200 (Sat, 16 May 2009) \$ (revision \$Rev: 6075 (2010-01-17) \$)

Since *date* indicates where functions or parameters have been added since the baseline of the API (TWiki release 4.2.3). The *date* indicates the earliest date of a Foswiki release that will support that function or parameter.

Deprecated *date* indicates where a function or parameters has been deprecated. Deprecated functions will still work, though they should *not* be called in new plugins and should be replaced in older plugins as soon as possible. Deprecated parameters are simply ignored in Foswiki releases after *date*.

Until *date* indicates where a function or parameter has been removed. The *date* indicates the latest date at which Foswiki releases still supported the function or parameter.

ClassMethod new(\$mode, \$user, \$web, \$topic, \$reason)

- \$mode - mode of access (view, change etc)
- \$user - canonical user name of user doing the accessing
- \$web - web being accessed
- \$topic - topic being accessed
- \$reason - string reason for failure

All the above fields are accessible from the object in a catch clause in the usual way e.g. `$e->{web}` and `$e->{reason}`

ObjectMethod stringify() -> \$string

Generate a summary string. This is mainly for debugging.

[Edit](#) | [Attach](#) | [Print version](#) | [History: %REVISIONS%](#) | [Backlinks](#) | [Raw View](#) | [More topic actions](#)

Topic revision: r1 - 12 Sep 2009 - 04:10:27 - ProjectContributor

- ☐ System

- Log In

- **Toolbox**

-  Users
-  Groups
-  Index
-  Search
-  Changes
-  Notifications
-  RSS Feed
-  Statistics
-  Preferences

- **User Reference**

- BeginnersStartHere
- TextFormattingRules
- Macros
- FormattedSearch
- QuerySearch
- DocumentGraphics
- SkinBrowser
- InstalledPlugins

- **Admin Maintenance**

- Reference Manual
- AdminToolsCategory
- InterWikis
- ManagingWebs
- SiteTools
- DefaultPreferences

- WebPreferences
- **Categories**
 - Admin Documentation
 - Admin Tools
 - Developer Doc
 - User Documentation
 - User Tools
- **Webs**
 - ☐ Public
 - ☐ System

-
-



Copyright © by the contributing authors. All material on this site is the property of the contributing authors.

Ideas, requests, problems regarding Wiki? Send feedback