

TYPO3.Flow - Task # 37451

Status:	Resolved	Priority:	Should have
Author:	Rens Admiraal	Category:	Security
Created:	2012-05-23	Assigned To:	Rens Admiraal
Updated:	2013-08-14	Due date:	
Sprint:			
PHP Version:			
Has patch:	No		
Complexity:			
Subject:	Rework usage of Roles in the security framework		
Description			
Roles are now simple string values but there's actually an entity object for it. We need the Roles to be persisted (like for example to add new roles in the phoenix backend)			
Related issues:			
related to TYPO3.Flow - Bug # 28862: Inconsistent handling of roles as object...		Resolved	2011-08-09

Associated revisions

Revision a79904d9 - 2012-05-23 22:04 - Rens Admiraal

[FEATURE] Add hasRole() method to Account entity

At the moment the roles still have to be persisted as strings in the serialized roles property in the database.

Already converting all roles to objects would require some more refactoring.

This change is tested on already existing, and on new users.

Change-Id: Ibb4989f43d4e92e2e19732786999654eb6531fb8

Releases: 1.2

Related: #37451

Revision f0776fb3 - 2013-04-10 17:25 - Karsten Dambekalns

[!!!][FEATURE] Handle roles consistently as objects

With this change roles are consistently used as objects.

To prevent naming conflicts between roles from different packages the package key is added to the role names.

The handling of roles now is more strict and does not allow the use of not configured (non-existent) roles anymore.

Roles are now referenced as real instances instead of their string identifiers in models. Roles which were defined in policies are now automatically mirrored to the RoleRepository (usually stored as a table in a database).

The getRoles() method has been removed from TokenInterface.

Change-Id: I39f335baa01fbe9b883e98e85cb1f063bbaf5744

Resolves: #28862

Resolves: #43578

Related: #37451

Related: #27698

Releases: 2.0, master

Revision 28e52aa8 - 2013-04-11 12:27 - Karsten Dambekalns

[!!!][FEATURE] Handle roles consistently as objects

With this change roles are consistently used as objects.

To prevent naming conflicts between roles from different packages the package key is added to the role names.

The handling of roles now is more strict and does not allow the use of not configured (non-existent) roles anymore.

Roles are now referenced as real instances instead of their string identifiers in models. Roles which were defined in policies are now automatically mirrored to the RoleRepository (usually stored as a table in a database).

The getRoles() method has been removed from TokenInterface.

Change-Id: I39f335baa01fbe9b883e98e85cb1f063bbaf5744

Resolves: #28862

Resolves: #43578

Related: #37451

Related: #27698

Releases: 2.0, master

History

#1 - 2012-05-24 09:41 - Karsten Dambekalns

- Project changed from TYPO3 Flow Base Distribution to TYPO3.Flow

#2 - 2012-05-24 09:41 - Karsten Dambekalns

- Category set to Security

- Has patch set to No

#3 - 2012-05-24 10:13 - Rens Admiraal

Some short notes from a discussion with Christian:

- Role is an entity
- Maybe we should keep special roles in memory only (like anonymous and everybody, which are both not assigned by relation but by context)
- Matching on string values should still be possible
- Roles configured in Policy.yaml should automatically be persisted (probably in the policyService)

- Identifier has to be unique, but should probably not be annotated @ORM\Id
- There should be an easy way to find all parent roles

#4 - 2012-08-02 08:59 - Robert Lemke

Rens Admiraal wrote:

- *Role is an entity*

yes!

- *Maybe we should keep special roles in memory only (like anonymous and everybody, which are both not assigned by relation but by context)*

yes, anonymous and everybody are hard-wired in the security framework and certainly should stay like that. So: don't persist.

- *Matching on string values should still be possible*

you mean hasRole(\$string) etc? Then sure!

- *Roles configured in Policy.yaml should automatically be persisted (probably in the policyService)*

And there we start mixing things up ... I think what you really want is persisting groups, not persisting roles. Let's discuss that.

- *Identifier has to be unique, but should probably not be annotated @ORM\Id*
- *There should be an easy way to find all parent roles*

#5 - 2012-08-19 22:58 - Rens Admiraal

- *Assigned To set to Rens Admiraal*

#6 - 2012-08-21 13:03 - Rens Admiraal

- *Subject changed from Store roles as an entity to Rework usage of Roles in the security framework*

The changes pushed to the review queue implement the following concept:

- Roles are defined in a programmatic way (Policy.yaml)
- The policyService / securityContext handle Roles as objects (not as string, exception: the hasRole requires a string representation)
- Roles reside in their own package namespace (like Foo.MyPackage.RoleName)
- 'Old style definition' should still work
- The framework should support nestable groups which can have roles assigned to make roles configuration more flexible in an application if needed

#7 - 2012-11-21 14:02 - Karsten Dambekalns

- *Status changed from New to Under Review*

See <https://review.typo3.org/#/q/topic:roles,n,z>

#8 - 2012-11-21 14:04 - Karsten Dambekalns

- *Target version set to 2.0*

What those changes are missing as of today - to fully replace the idea of the AuthorizationGroup - is an addRole() method

#9 - 2012-12-09 21:25 - Karsten Dambekalns

- *Target version changed from 2.0 to 2.1*

#10 - 2013-05-21 13:28 - Robert Lemke

- *Target version deleted (2.1)*

#11 - 2013-08-14 15:59 - Karsten Dambekalns

- *Status changed from Under Review to Resolved*

- *Target version set to 2.0*

- *% Done changed from 0 to 100*