— tech du peupl

Language : English Français

Share

[4/4] Symfony 2.3, SonataAdminBundle, FOSUserBundle and ACLs

Part 4 of 4.

Now that our setup is complete, let's try to actually use these ACLs in SonataAdminBundle dashboard.

Let's see the summary one last time, in case you missed a previous step :

1. Symfony 2.3 : ACL, SonataAdminBundle, SonataMediabundle, SonataUserBundle and FOSUserBundle
2. Symfony 2.3, SonataAdminBundle, SonataMediabundle, SonataUserBundle and FOSUserBundle : a good project base
3. Symfony 2.3 : Customizing SonataAdminBundle
4. Symfony 2.3, SonataAdminBundle, FOSUserBundle and ACLs

Setting up permissions and authorizations in SonataAdminBundle can be done the same as with any Symfony2 application : either with roles or with ACLs. Roles allow to specify which actions can be performed by users assignated to a specific role. ACLs is a concept where permissions can be set for tuple (user, action, object). By example : userA can perform actionB on objectC. Since ACLs allow fine-grain access controls, we'll choose this system. One word of caution though : Sonata ACLs are immutable.

First, we will have to prepare our groups by extending SonataUser defaults :
/src/Tech/NewsBundle/Entity/Group.php

```php
1  < ?php
2
3  namespace Tech\NewsBundle\Entity;
4
5  use Sonata\UserBundle\Entity\BaseGroup as BaseGroup;
6  use Doctrine\ORM\Mapping as ORM;
7
8  /**
9   * @ORM\Entity
```

```
 9        
10    * @ORM\Table(name="fos_group")
11    */
12   class Group extends BaseGroup
13   {
14       /**
15        * @ORM\Id
16        * @ORM\Column(type="integer")
17        * @ORM\GeneratedValue(strategy="AUTO")
18        */
19       protected $id;
20   }
```

↬ Other blogs you might like !

Categories

config.yml : fos_user section : Two lines to add, one to uncomment. sonata_user section :

one line to add.

```
 1   fos_user:
 2       db_driver: orm
 3       firewall_name: main
 4       user_class: Tech\NewsBundle\Entity\User
 5       group:
 6           group_class:  Tech\NewsBundle\Entity\Group
 7       profile:  # Authentication Form
 8           form:
 9               type:           fos_user_profile
10               handler:        fos_user.profile.form.handler.defau
11               name:           fos_user_profile_form
12               #validation_groups:  [Authentication] # Please note : thi
13   [...]
14   sonata_user:
15       security_acl:     false
16       class:
17           user:         Tech\NewsBundle\Entity\User
18           group:        Tech\NewsBundle\Entity\Group
19           [...]
```

Then update the database schema : ⚡ php app/console doctrine:schema:update --

force

It's time to activate the ACLs and populate the database with some initial rules and objects.

We have to add a new section to **security.yml** :

```
1   parameters:
2       security.acl.permission.map.class: Sonata\AdminBundle\Security\A
```

config.yml :

```
 1   sonata_admin:
 2       [...]
 3       security:
 4           handler: sonata.admin.security.handler.acl
 5
 6           # acl security information
 7           information:
 8               GUEST:    [VIEW, LIST]
 9               STAFF:    [EDIT, LIST, CREATE]
10               EDITOR:   [OPERATOR, EXPORT]
11               ADMIN:    [MASTER]
12           # permissions not related to an object instance and also to
13           # the DELETE admin permission means the user is allowed to b
14           admin_permissions: [CREATE, LIST, DELETE, UNDELETE, EXPORT,
15           # permission related to the objects
16           object_permissions: [VIEW, EDIT, DELETE, UNDELETE, OPERATOR,
17   [...]
18   sonata_user:
19       security_acl:     true
```

Then back to the terminal to run some commands :

⚘ `php app/console init:acl`

⚘ `php app/console sonata:admin:generate-object-acl`

⚘ `php app/console sonata:admin:setup-acl`

Now, if you go back to your dashboard, you should notice in Users > Groups that several roles have been created. Create a new group, name it Staff, and check the NEWS and MEDIA staff roles and the ROLE_ADMIN role. You should then be able to add users to this group. As stated in the documentation, they will be able to EDIT, LIST and CREATE news.

Create a staff user, assign it to the staff group, sign in using this new user and browse the news list. You'll be able to edit the news. When you create a news, you'll be automatically assigned the OWNER permission on this news, which includes DELETE permission. Sonata documentation (see link above) will tell you more about existing ACLs and their specifics.

This post is long enough, I could add one more (and possibly last since it'll be the fifth of a four-serie) depending on my time and progress. We'll see.

Tags :

4 comments

★ ‹ 0

Leave a message...

Best ▾   **Community**   **Share** ⬆   ⚙▾

no name • 2 months ago

Thank you for this serie

1 ⌃ | ⌄   Reply   Share ›

Guest • a month ago

Thanks for the series, it's been very helpful. Everything has worked for me perfectly up to updating the database schema with the ACLs. After trying to update the database I get the following error (that I'm hoping you can point me in the right direction to fix it)

[Doctrine\DBAL\DBALException] An exception occurred while executing 'ALTER TABLE fos_user_user_group ADD CONSTRAINT FK_B3XXXXXXX47 FOREIGN KEY (group_id) REFERENCES fos_group (id) ON DELETE CASCADE':

SQLSTATE[23000]: Integrity constraint violation: 1452 Cannot add or update a child row: a foreign key constraint fails (`symfony_db`.`#sql-48e_ea`, CONSTRAINT `FK_B3XXXXXXX47` FOREIGN KEY (`group_id`) REFERENCES `fos_group` (`id`) ON DELETE CASCADE)

[PDOException] SQLSTATE[23000]: Integrity constraint violation: 1452 Cannot add or update a child row: a foreign key constraint fails (`symfony_db`.`#sql-48e_ea`, CONSTRAINT `FK_B3XXXXXXX47` FOREIGN KEY (`group_id`) REFERENCES `fos_group` (`id`) ON DELETE CASCADE)

⌃ | ⌄   Reply   Share ›

Bob Door ➔ Guest • 25 days ago

Nevermind, I just dropped the table and rebuilt it. Works fine!

⌃ | ⌄   Reply   Share ›

Sylvain • 2 months ago

J'attends avec impatience pour la suite avec la partie 4 :)

⌃ | ⌄   Reply   Share ›