



eBook Gratuit

APPRENEZ sonata-admin

eBook gratuit non affilié créé à partir des
contributeurs de Stack Overflow.

#sonata-
admin

Table des matières

À propos	1
Chapitre 1: Démarrer avec sonata-admin	2
Remarques.....	2
Exemples.....	2
Installation ou configuration.....	2
Chapitre 2: Intégrez vos électeurs de sécurité avec sonata-admin	3
Introduction.....	3
Exemples.....	3
Mettre tout en place.....	3
Crédits	7

À propos

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [sonata-admin](#)

It is an unofficial and free sonata-admin ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official sonata-admin.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapitre 1: Démarrer avec sonata-admin

Remarques

Cette section fournit une vue d'ensemble de ce qu'est sonata-admin et pourquoi un développeur peut vouloir l'utiliser.

Il doit également mentionner tous les sujets importants dans sonata-admin, et établir un lien avec les sujets connexes. La documentation de sonata-admin étant nouvelle, vous devrez peut-être créer des versions initiales de ces rubriques connexes.

Exemples

Installation ou configuration

Instructions détaillées sur la configuration ou l'installation de sonata-admin.

Lire **Démarrer avec sonata-admin en ligne**: <https://riptutorial.com/fr/sonata-admin/topic/8094/demarrer-avec-sonata-admin>

Chapitre 2: Intégrez vos électeurs de sécurité avec sonata-admin.

Introduction

Lors de la configuration de voteurs à utiliser avec sonata-admin, il y a quelques pièges, les étapes présentées ici doivent vous permettre d'être opérationnel, afin que la sonate prenne correctement en compte les votants, lors du rendu des boutons d'édition, la barre latérale, lors de l'exécution des actions par lots, fondamentalement, il fonctionnera finalement comme vous le souhaitiez en premier lieu ..

Exemples

Mettre tout en place

Premièrement, nous avons créé un électeur:

```
namespace BBIT\CoreBundle\Security\Authorization\Voter;

use Symfony\Component\Security\Core\Authentication\Token\TokenInterface;
use Symfony\Component\Security\Core\Authorization\Voter\VoterInterface;

class EventVoter implements VoterInterface
{
    const VIEW      = 'VIEW';
    const EDIT      = 'EDIT';
    const DELETE    = 'DELETE';
    const CREATE    = 'CREATE';
    const LLIST     = 'LLIST';

    public function supportsAttribute($attribute)
    {
        return in_array($attribute, array(
            self::VIEW,
            self::EDIT,
            self::DELETE,
            self::CREATE,
            self::LLIST,
        ));
    }

    public function supportsClass($class)
    {
        $supportedClass = 'BBIT\CoreBundle\Entity\SomeEntity';

        if (is_string($class)) {
            if ($class === $supportedClass) {
                return true;
            } else {
```

```

        return false;
    }
}
return $supportedClass === get_class($class) || is_subclass_of(get_class($class),
$supportedClass);
}

public function vote(TokenInterface $token, $entity, array $attributes)
{
    $user = $token->getUser();

    if (!is_object($user)) {
        return VoterInterface::ACCESS_DENIED;
    }

    if (!$this->supportsClass($entity)) {
        return VoterInterface::ACCESS_ABSTAIN;
    }

    $attribute = $attributes[0];

    switch($attribute) {
        case self::LLIST:

            return VoterInterface::ACCESS_DENIED;
            break;
        case self::VIEW:

            return VoterInterface::ACCESS_DENIED;
            break;
        case self::CREATE:

            return VoterInterface::ACCESS_DENIED;
            break;
        case self::EDIT:

            return VoterInterface::ACCESS_DENIED;
            break;
        case self::DELETE:

            return VoterInterface::ACCESS_DENIED;
            break;
    }

    return VoterInterface::ACCESS_DENIED;
}
}
}

```

Cet électeur est légèrement différent de l'électeur par défaut dans les documents Symfony, avec l'avantage supplémentaire de pouvoir accepter un objet, ou le nom de classe lui-même, en tant qu'argument.

Deuxièmement, nous allons créer un VoterSecurityhandler, qui étend et remplace une partie du RoleSecurityHandler de sonata:

```

namespace BBIT\CoreBundle\Security\Handler;

use Sonata\AdminBundle\Admin\AdminInterface;

```

```

use Sonata\AdminBundle\Security\Handler\RoleSecurityHandler;
use Symfony\Component\Security\Core\Exception\AuthenticationCredentialsNotFoundException;

class VoterSecurityHandler extends RoleSecurityHandler
{

    /**
     * {@inheritdoc}
     */
    public function isGranted(AdminInterface $admin, $attributes, $object = null)
    {
        if (!is_array($attributes)) {
            $attributes = array($attributes);
        }

        if ($object == $admin) {
            $object = $admin->getClass();
        }

        foreach ($attributes as $pos => $attribute) {
            $attributes[$pos] = $attribute;
        }

        try {
            return $this->securityContext->isGranted($attributes, $object);
        } catch (AuthenticationCredentialsNotFoundException $e) {
            return false;
        } catch (\Exception $e) {
            throw $e;
        }
    }

    /**
     * {@inheritdoc}
     */
    public function getBaseRole(AdminInterface $admin)
    {
        return '%s';
    }
}

```

Maintenant, nous avons besoin d'une définition de service pour définir ce gestionnaire en tant que service:

```

services:
    ...
    sonata.admin.security.handler.voter:
        class: BBIT\CoreBundle\Security\Handler\VoterSecurityHandler
        arguments:
            - @security.context
            - [ROLE_SUPER_ADMIN]

```

Maintenant, nous pouvons dire à la sonate d'utiliser notre VoterSecurityHandler:

```

sonata_admin:

```

```
...  
security:  
  handler: sonata.admin.security.handler.voter
```

That's it at this point, the sonata will take account of your voter, and you should be good to go.

[Lire Intégrez vos électeurs de sécurité avec sonata-admin. en ligne:](https://riptutorial.com/fr/sonata-admin/topic/10030/integrez-vos-electeurs-de-securite-avec-sonata-admin-)

<https://riptutorial.com/fr/sonata-admin/topic/10030/integrez-vos-electeurs-de-securite-avec-sonata-admin->

Crédits

S. No	Chapitres	Contributeurs
1	Démarrer avec sonata-admin	Community
2	Intégrez vos électeurs de sécurité avec sonata-admin.	Sam Janssens