



FREE eBook

LEARNING sonata-admin

Free unaffiliated eBook created from
Stack Overflow contributors.

#sonata-
admin

Table of Contents

About	1
Chapter 1: Getting started with sonata-admin	2
Remarks.....	2
Examples.....	2
Installation or Setup.....	2
Chapter 2: Integrate your security voters with sonata-admin	3
Introduction.....	3
Examples.....	3
Setting it all up.....	3
Credits	7

About

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

Chapter 1: Getting started with sonata-admin

Remarks

This section provides an overview of what sonata-admin is, and why a developer might want to use it.

It should also mention any large subjects within sonata-admin, and link out to the related topics. Since the Documentation for sonata-admin is new, you may need to create initial versions of those related topics.

Examples

Installation or Setup

Detailed instructions on getting sonata-admin set up or installed.

Read *Getting started with sonata-admin* online: <https://riptutorial.com/sonata-admin/topic/8094/getting-started-with-sonata-admin>

Chapter 2: Integrate your security voters with sonata-admin.

Introduction

When setting up voters to use with sonata-admin, there are a few pitfalls, the steps shown here should get you up and running, so that sonata properly takes the voters into account, when rendering the edit, show and delete buttons, when building the sidebar, when running the batch actions, basically, it will finally work the way you expected it to work in the first place ..

Examples

Setting it all up

Firstly, we created a voter:

```
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;
}
}
}

```

This voter is slightly different from the default voter in the symfony docs, with the added benefit of being able to accept either an object, or the classname itself, as an argument.

Secondly, we are going to create a VoterSecurityhandler, wich extends from, and overwrites part of, sonata's RoleSecurityHandler:

```

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';
    }
}

```

Now we need a service-definition to define this handler as a service:

```

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

```

Now we can tell sonata to use our VoterSecurityHandler:

```

sonata_admin:
    ...
    security:
        handler: sonata.admin.security.handler.voter

```

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

Read [Integrate your security voters with sonata-admin](https://riptutorial.com/sonata-admin/topic/10030/integrate-your-security-voters-with-sonata-admin). online: [https://riptutorial.com/sonata-admin/topic/10030/integrate-your-security-voters-with-sonata-admin-](https://riptutorial.com/sonata-admin/topic/10030/integrate-your-security-voters-with-sonata-admin)

Credits

S. No	Chapters	Contributors
1	Getting started with sonata-admin	Community
2	Integrate your security voters with sonata-admin.	Sam Janssens