



EBook Gratis

APRENDIZAJE sonata-admin

Free unaffiliated eBook created from
Stack Overflow contributors.

#sonata-
admin

Tabla de contenido

Acerca de.....	1
Capítulo 1: Empezando con sonata-admin.....	2
Observaciones.....	2
Examples.....	2
Instalación o configuración.....	2
Capítulo 2: Integrar sus votantes de seguridad con sonata-admin.....	3
Introducción.....	3
Examples.....	3
Configurando todo.....	3
Creditos.....	7

Acerca de

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

Capítulo 1: Empezando con sonata-admin

Observaciones

Esta sección proporciona una descripción general de qué es sonata-admin y por qué un desarrollador puede querer usarla.

También debe mencionar cualquier tema importante dentro de sonata-admin y vincular a los temas relacionados. Dado que la Documentación para sonata-admin es nueva, es posible que deba crear versiones iniciales de los temas relacionados.

Examples

Instalación o configuración

Instrucciones detalladas sobre cómo configurar o instalar sonata-admin.

Lea *Empezando con sonata-admin en línea*: <https://riptutorial.com/es/sonata-admin/topic/8094/empezando-con-sonata-admin>

Capítulo 2: Integrar sus votantes de seguridad con sonata-admin.

Introducción

Cuando configura a los votantes para que usen con sonata-admin, existen algunos escollos, los pasos que se muestran aquí deberían ayudarlo a ponerse en marcha, de modo que la sonata tome en cuenta a los votantes de manera adecuada, al renderizar, mostrar y eliminar los botones, al crear la barra lateral, cuando ejecuta las acciones por lotes, básicamente, finalmente funcionará de la manera que esperaba que funcionara en primer lugar ..

Examples

Configurando todo

En primer lugar, creamos un votante:

```
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;
}
}
}

```

Este votante es ligeramente diferente del votante predeterminado en los documentos de Symfony, con el beneficio adicional de ser capaz de aceptar un objeto, o el propio nombre de clase, como un argumento.

En segundo lugar, vamos a crear un Manejador de Seguridad de Votante, que se extiende y sobrescribe parte de, 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';
    }
}

```

Ahora necesitamos una definición de servicio para definir este controlador como un servicio:

```

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

```

Ahora podemos decirle a la sonata que use nuestro Gestor de seguridad para votantes:

```

sonata_admin:
    ...
    security:

```

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

Eso es, en este punto, la sonata tomará en cuenta a su votante, y debería estar listo.

Lea [Integrar sus votantes de seguridad con sonata-admin](https://riptutorial.com/es/sonata-admin/topic/10030/integrar-sus-votantes-de-seguridad-con-sonata-admin-). en línea:

<https://riptutorial.com/es/sonata-admin/topic/10030/integrar-sus-votantes-de-seguridad-con-sonata-admin->

Creditos

S. No	Capítulos	Contributors
1	Empezando con sonata-admin	Community
2	Integrar sus votantes de seguridad con sonata-admin.	Sam Janssens