SYMFONY2

High Performance PHP Framework for Web Development

Symfony is a PHP framework for web projects. Speed up the creation and maintenance of your PHP web applications. Replace the repetitive coding tasks by power, control and pleasure.

Quang Tran

Sutrix Media (Vietnam) JSC.

CONTENT OWNER:

Sutrix Media

All future revisions to this document shall be approved by the content owner prior to release.

HCMC Office: 72/4 Truong Quoc Dung,
District Phu Nhuan, HCMC, Viet Nam
☏: +84 8 39975901 🖷: +84 8 3997 5900
HK Office: Unit 706, 7/F., South Seas, Towers,
75 Mody Road, TsimShaTsui, Hong Kong
☏: +852 8197 1217
🖱: **www.sutrixmedia.com**

The Sutrixmedia Joint Stock Company.

1. Signature

Originator By: Quang Tran	Date: 27/01/2014
Prepared By:	
Approved By: Thuan Nguyen	Date: 21/03/2014
Reviewed By:	
Distributed To:	

HCMC Office: 72/4 Truong Quoc Dung,
District Phu Nhuan, HCMC, Viet Nam
☏: +84 8 39975901 🖷: +84 8 3997 5900
HK Office: Unit 706, 7/F., South Seas, Towers,
75 Mody Road, TsimShaTsui, Hong Kong
☏: +852 8197 1217
🖱: **www.sutrixmedia.com**

2. Revision History

***A – Added, M – Modified, D - Deleted**

Version	Date	A*, M, D	Change Description	Author	Approved By
1.0	27.01.2014	A*		Quang Tran	

HCMC Office: 72/4 Truong Quoc Dung,
District Phu Nhuan, HCMC, Viet Nam
☏: +84 8 39975901 🖷: +84 8 3997 5900
HK Office: Unit 706, 7/F., South Seas, Towers,
75 Mody Road, TsimShaTsui, Hong Kong
☏: +852 8197 1217
🖱: **www.sutrixmedia.com**

3. Table of Contents

HCMC Office: 72/4 Truong Quoc Dung,
District Phu Nhuan, HCMC, Viet Nam
☏: +84 8 39975901 🖷: +84 8 3997 5900
HK Office: Unit 706, 7/F., South Seas, Towers,
75 Mody Road, TsimShaTsui, Hong Kong
☏: +852 8197 1217
🖱: **www.sutrixmedia.com**

4. Overview

Symfony is a PHP framework for web projects. Speed up the creation and maintenance of your PHP web applications. Replace the repetitive coding tasks by power, control and pleasure.

HCMC Office: 72/4 Truong Quoc Dung,
District Phu Nhuan, HCMC, Viet Nam
☏: +84 8 39975901 🖷: +84 8 3997 5900
HK Office: Unit 706, 7/F., South Seas, Towers,
75 Mody Road, TsimShaTsui, Hong Kong
☏: +852 8197 1217
🖱: **www.sutrixmedia.com**

5. Installing Symfony

a) Install symfony with composer.phar:

– Go to http://getcomposer.org/download/ to download composer.phar. Copy it to folder you want install and run command line at that folder:

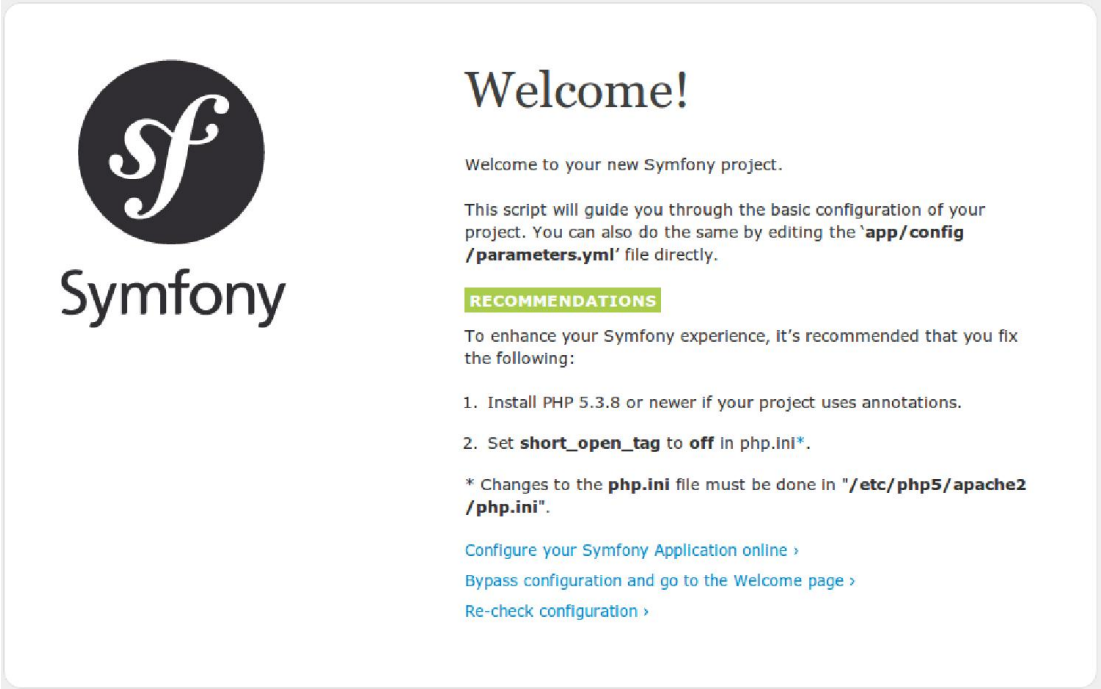
```
   php composer.phar create-project symfony/framework-standard-
edition symfony 2.1.4
```

b) Go to home page of symfony (http://symfony.com/download) and download Symfony Standard Edtion 2.1.4 (.zip). Extract it to folder you want install.

Note: Compatibility check your server with symfony 2.1:

Go to :

```
   http://localhost/symfony_path/web/config.php
```



HCMC Office: 72/4 Truong Quoc Dung, District Phu Nhuan, HCMC, Viet Nam
☏: +84 8 39975901 🖷: +84 8 3997 5900
HK Office: Unit 706, 7/F., South Seas, Towers, 75 Mody Road, TsimShaTsui, Hong Kong
☏: +852 8197 1217
🖱: www.sutrixmedia.com

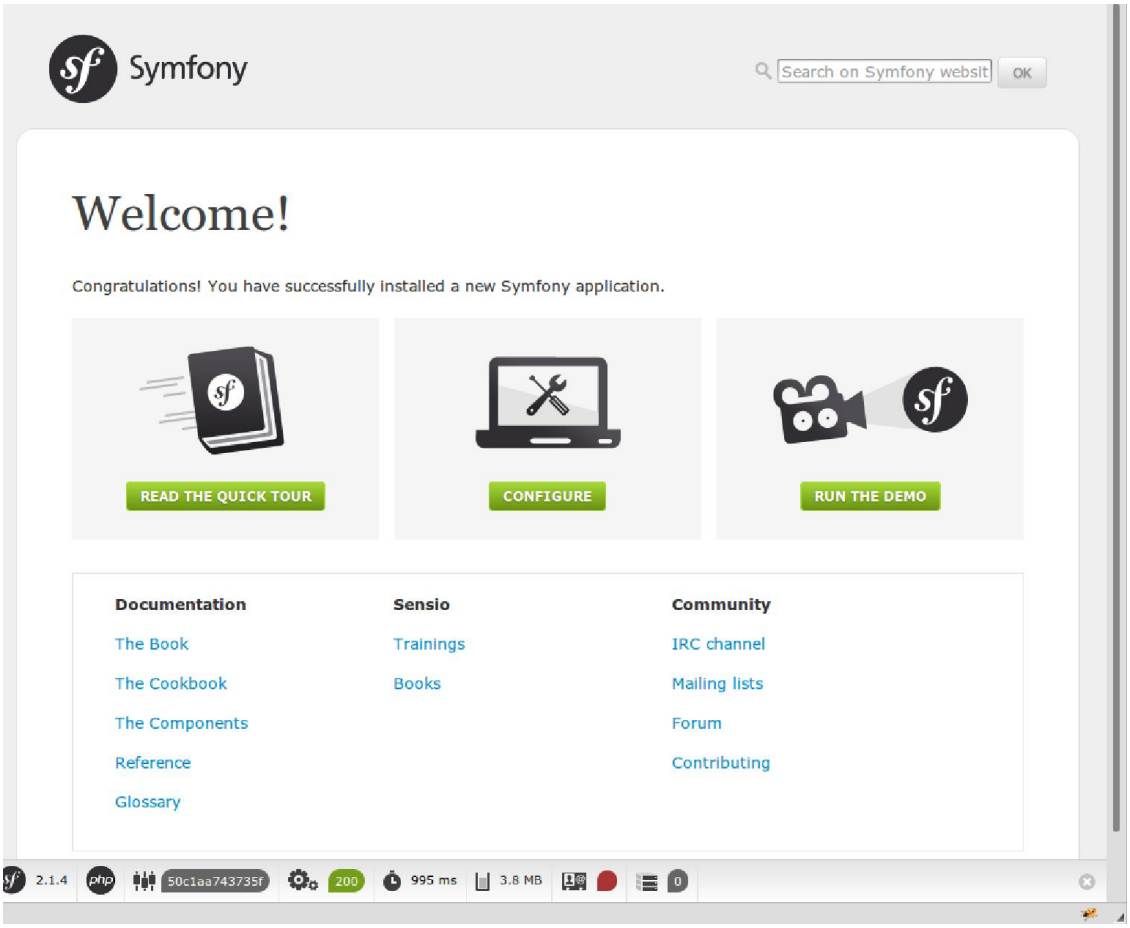
This page show for you how to enhance your Symfony experience, it's recommended that you fix your server.

c) Config symfony work with database:

– When you completed download symfony, go to:

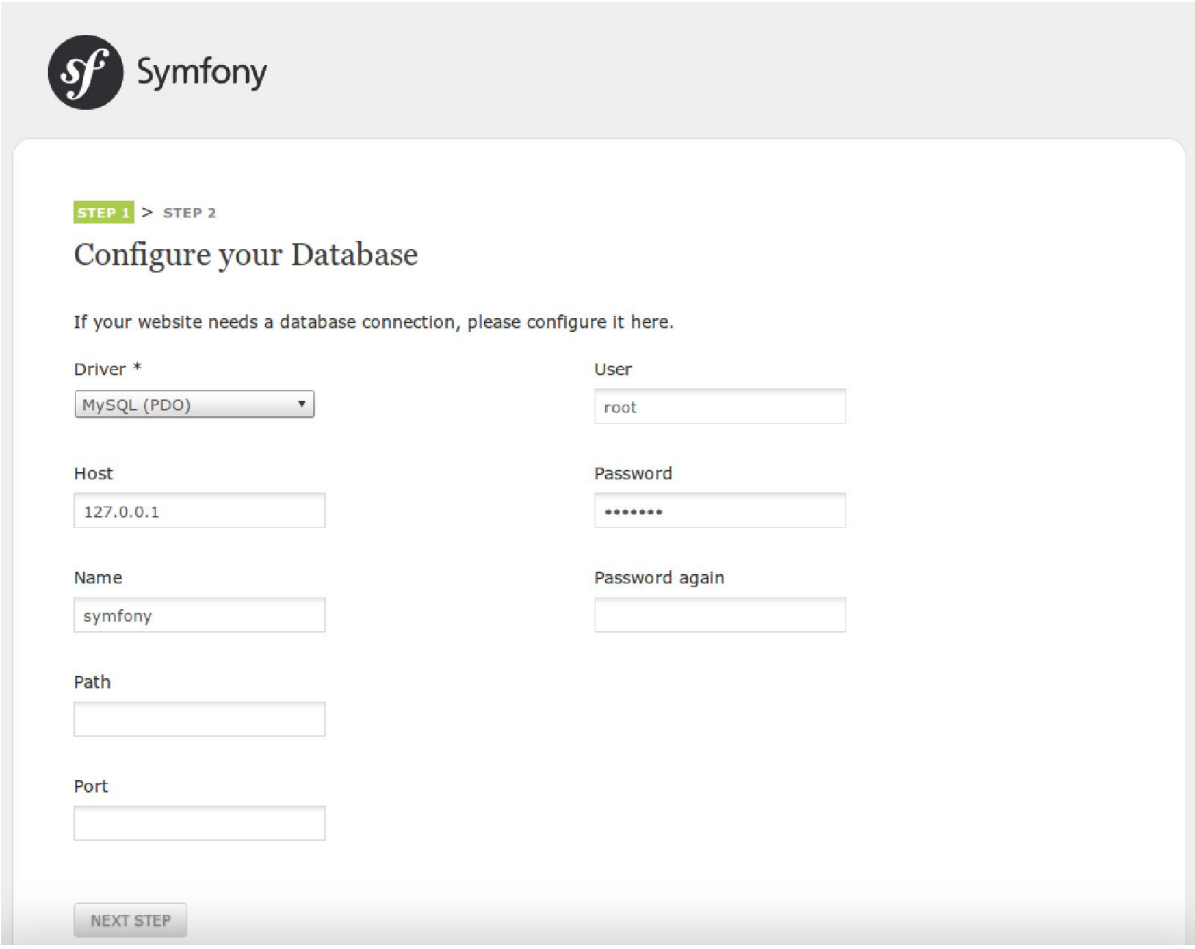
```
http://localhost/symfony_path/web/app_dev.php/
```

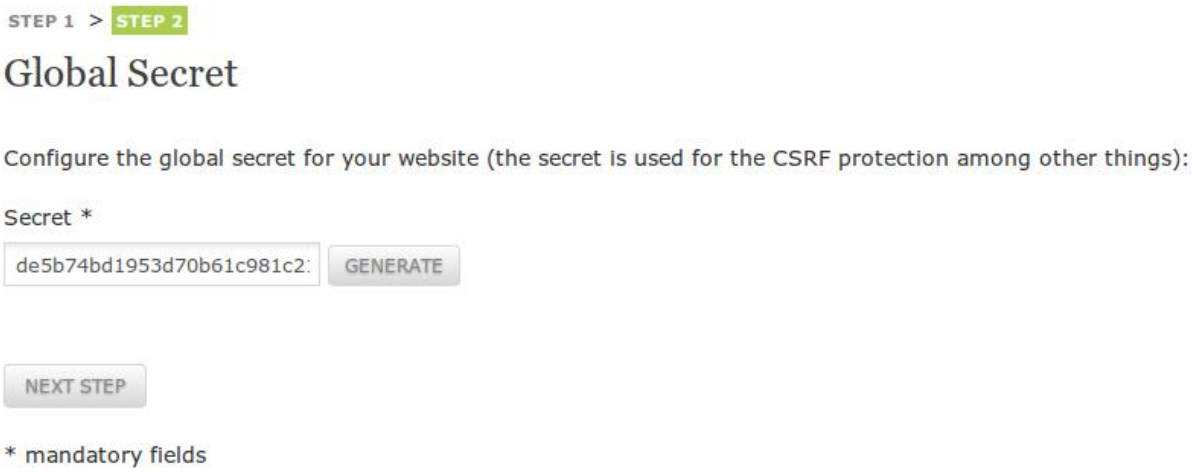
– Click to configure to config symfony in your server:



Follow with screen to config your database. Click next step to set up your Global Secret

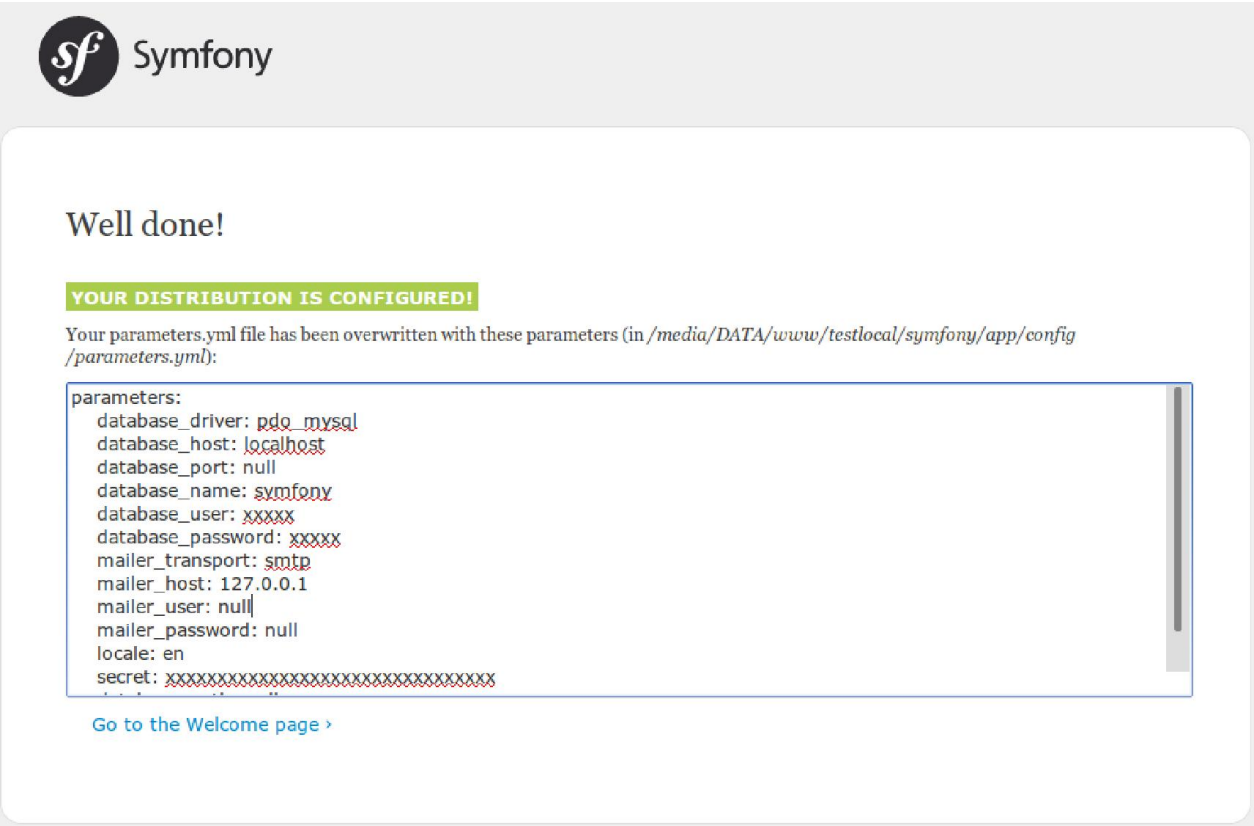
HCMC Office: 72/4 Truong Quoc Dung, District Phu Nhuan, HCMC, Viet Nam
☏: +84 8 39975901 🖷: +84 8 3997 5900
HK Office: Unit 706, 7/F., South Seas, Towers, 75 Mody Road, TsimShaTsui, Hong Kong
☏: +852 8197 1217
: www.sutrixmedia.com





HCMC Office: 72/4 Truong Quoc Dung, District Phu Nhuan, HCMC, Viet Nam
☏: +84 8 39975901 🖷: +84 8 3997 5900
HK Office: Unit 706, 7/F., South Seas, Towers, 75 Mody Road, TsimShaTsui, Hong Kong
☏: +852 8197 1217
: www.sutrixmedia.com

If done you will see



HCMC Office: 72/4 Truong Quoc Dung, District Phu Nhuan, HCMC, Viet Nam
☏: +84 8 39975901 🖷: +84 8 3997 5900
HK Office: Unit 706, 7/F., South Seas, Towers, 75 Mody Road, TsimShaTsui, Hong Kong
☏: +852 8197 1217
🖱: **www.sutrixmedia.com**

6. Installing Sonata Admin

a) Have many ways to install Sonata Admin. You can manual download it and coppy to vendor folders and config app/AppKernel.php to register this bundle. But for best, I recommend you install it though composer.phar. In this document. I will show for you how to install Sonata Admin also another bundle.

b) In 1.1 download composer.phar and copy it to folder of symfony (same level with folder app, web, …)

c) To setup Sonata Admin, you need install some bundle dependency with sonata. You can find requires in

```
https://packagist.org/packages/sonata-project/admin-bundle
```

In folder of symfony, open composer.json add:

```
    "require": {
        "php": ">=5.3.3",
        [...]
        "knplabs/knp-menu": "2.0.x-dev"
    },
```

Go to command line at folder of symfony, run command:

```
php composer.phar update
```

If success, you will see result:

```
- Installing knplabs/knp-menu (dev-master 335bb6f)
  Cloning 335bb6f12c589fbfd32a2de37d248262148c3a62
```

Open again composer.json, remove line "knplabs/knp-menu": "2.0.x-dev" and add:

HCMC Office: 72/4 Truong Quoc Dung, District Phu Nhuan, HCMC, Viet Nam
☏: +84 8 39975901 🖷: +84 8 3997 5900
HK Office: Unit 706, 7/F., South Seas, Towers, 75 Mody Road, TsimShaTsui, Hong Kong
☏: +852 8197 1217
🖱: www.sutrixmedia.com

```
        "require": {
            "php": ">=5.3.3",
            [...]
           "sonata-project/admin-bundle": "dev-master",
            "sonata-project/jquery-bundle": "dev-master",
            "sonata-project/doctrine-orm-admin-bundle": "dev-master",
            "sonata-project/block-bundle": "dev-master",
            "sonata-project/exporter": "dev-master",
            "sonata-project/cache-bundle": "dev-master",
            "sonata-project/intl-bundle": "dev-master",
            "sonata-project/user-bundle": "dev-master",
            "friendsofsymfony/user-bundle": "2.0.x-dev",
            "willdurand/propel-typehintable-behavior": "dev-master",
            "sonata-project/doctrine-extensions": "dev-master",
            "sonata-project/easy-extends-bundle": "dev-master",
            "sonata-project/google-authenticator": "dev-master"
        },
```

Note: to install intl-bundle you must enable php extension intl

If you success, you will see this result:

```
Loading composer repositories with package information
Updating dependencies
  - Installing sonata-project/jquery-bundle (dev-master 5f87a76)
    Cloning 5f87a761302e6c78304e071416f75d41eb1fb3c2

  - Updating knplabs/knp-menu (dev-master 335bb6f => v1.1.2)
    Checking out v1.1.2

  - Installing knplabs/knp-menu-bundle (v1.1.0)
    Downloading: 100%

  - Installing sonata-project/block-bundle (dev-master 60b12fd)
    Cloning 60b12fd035b8f2e68f1ead4800dd4e6253800028

  - Installing sonata-project/exporter (dev-master 75174b5)
    Cloning 75174b5a41f1014ecb4b4810a518f036b830d862

  - Installing sonata-project/admin-bundle (dev-master 57ef26f)
    Cloning 57ef26fe378643d3cecc16c65b270e446d19c8a7
```

Config Sonata admin

HCMC Office: 72/4 Truong Quoc Dung, District Phu Nhuan, HCMC, Viet Nam
☏: +84 8 39975901 🖷: +84 8 3997 5900
HK Office: Unit 706, 7/F., South Seas, Towers, 75 Mody Road, TsimShaTsui, Hong Kong
☏: +852 8197 1217
🖱: www.sutrixmedia.com

After install, be sure to enable this bundels in your AppKernel.php file:

```
# app/AppKernel.php
public function registerBundles()
{
    return array(
        // ...
        new Sonata\UserBundle\SonataUserBundle('FOSUserBundle'),
        new Sonata\BlockBundle\SonataBlockBundle(),
        new Sonata\CacheBundle\SonataCacheBundle(),
        new Sonata\jQueryBundle\SonatajQueryBundle(),
        new Sonata\AdminBundle\SonataAdminBundle(),
        new FOS\UserBundle\FOSUserBundle(),
        new
Sonata\DoctrineORMAdminBundle\SonataDoctrineORMAdminBundle(),
        new Knp\Bundle\MenuBundle\KnpMenuBundle(),
        // ...
    );
}
```

Create new file app/configs/sonata.yml to config Sonata:

```
# app/config/sonata.yml
sonata_block:
    default_contexts: [cms]
    blocks:
        sonata.admin.block.admin_list:
            contexts:   [admin]

        sonata.block.service.text:
        sonata.block.service.action:
        sonata.block.service.rss:
```

Include this file to enable config:

HCMC Office: 72/4 Truong Quoc Dung, District Phu Nhuan, HCMC, Viet Nam
☏: +84 8 39975901 🖷: +84 8 3997 5900
HK Office: Unit 706, 7/F., South Seas, Towers, 75 Mody Road, TsimShaTsui, Hong Kong
☏: +852 8197 1217
🖱: www.sutrixmedia.com

```
# app/config/config.yml
imports:
    # ...
    - { resource: sonata.yml }
```

Create new application to extends SonataUser, in document I using namspace MS:

```
  php app/console generate:bundle --
namespace=MS/Bundle/UserBundle --format=yml
```

Open Ms\Bundle\UserBundle\MSUserBundle.php and edit again:

```
<?php
# src\Ms\Bundle\UserBundle\MSUserBundle.php

namespace MS\Bundle\UserBundle;

use Symfony\Component\HttpKernel\Bundle\Bundle;

class MSUserBundle extends Bundle
{
    public function getParent()
  {
        return 'SonataUserBundle';
  }
}
```

Create new Entity for this application:

```
  <?php
# src\ MS\Bundle\UserBundle\Entity\User.php

namespace MS\Bundle\UserBundle\Entity;

use Doctrine\ORM\Mapping as ORM;
use Sonata\UserBundle\Entity\BaseUser;
```


HCMC Office: 72/4 Truong Quoc Dung, District Phu Nhuan, HCMC, Viet Nam
☏: +84 8 39975901 🖷: +84 8 3997 5900
HK Office: Unit 706, 7/F., South Seas, Towers, 75 Mody Road, TsimShaTsui, Hong Kong
☏: +852 8197 1217
🖱: www.sutrixmedia.com

```
/**
 * MS\Bundle\UserBundle\Entity\User
 *
 * @ORM\Table(name="ms_user")
 * @ORM\Entity
 */
class User extends BaseUser
{
    /**
     * @var integer
     *
     * @ORM\Column(name="id", type="integer")
     * @ORM\Id
     * @ORM\GeneratedValue(strategy="AUTO")
     */
    protected $id;


    /**
     * Get id
     *
     * @return integer
     */
    public function getId()
    {
        return $this->id;
    }
}
```

You will also need to alter your app/config/config.yml file :

```
# app/config/config.yml
fos_user:
    db_driver:      orm # can be orm or odm
    firewall_name:  main
    user_class:     MS\Bundle\UserBundle\Entity\User
```


HCMC Office: 72/4 Truong Quoc Dung, District Phu Nhuan, HCMC, Viet Nam
☏: +84 8 39975901 🖷: +84 8 3997 5900
HK Office: Unit 706, 7/F., South Seas, Towers, 75 Mody Road, TsimShaTsui, Hong Kong
☏: +852 8197 1217
: www.sutrixmedia.com

```
     Config routing for sonata admin:

   # app/config/routing.yml
   # …
   admin:
       resource:
'@SonataAdminBundle/Resources/config/routing/sonata_admin.xml'
       prefix: /admin

   _sonata_admin:
       resource: .
       type: sonata_admin
       prefix: /admin

   sonata_user:
       resource:
'@SonataUserBundle/Resources/config/routing/admin_security.xml'
       prefix: /admin
```

At this point you have basic administration for your model. If you wish to quickly customize your administration you can create some configuration options and change them according to your requirements:

```
   # app/config/sonata.yml
   sonata_admin:
       title:      Sonata Project
       title_logo: /bundles/sonataadmin/logo_title.png
       templates:
           # default global templates
           layout:  SonataAdminBundle::standard_layout.html.twig
           ajax:    SonataAdminBundle::ajax_layout.html.twig

           # default actions templates, should extend a global
templates
           list:    SonataAdminBundle:CRUD:list.html.twig
           show:    SonataAdminBundle:CRUD:show.html.twig
           edit:    SonataAdminBundle:CRUD:edit.html.twig
       dashboard:
           blocks:
```


HCMC Office: 72/4 Truong Quoc Dung, District Phu Nhuan, HCMC, Viet Nam
☏: +84 8 39975901 🖷: +84 8 3997 5900
HK Office: Unit 706, 7/F., South Seas, Towers, 75 Mody Road, TsimShaTsui, Hong Kong
☏: +852 8197 1217
🖱: www.sutrixmedia.com

```
                # display a dashboard block
                - { position: left, type:
sonata.admin.block.admin_list }

    sonata_user:
        security_acl: true
        class:
          user:  MS\Bundle\UserBundle\Entity\User
    #      group: MS\Bundle\UserBundle\Entity\Group

    sonata_doctrine_orm_admin:
        # default value is null, so doctrine uses the value defined
in the configuration
        entity_manager: ~

        templates:
            form:
                -
SonataDoctrineORMAdminBundle:Form:form_admin_fields.html.twig
            filter:
                -
SonataDoctrineORMAdminBundle:Form:filter_admin_fields.html.twig
            types:
                list:
                    array:
SonataAdminBundle:CRUD:list_array.html.twig
                    boolean:
SonataAdminBundle:CRUD:list_boolean.html.twig
                    date:
SonataAdminBundle:CRUD:list_date.html.twig
                    time:
SonataAdminBundle:CRUD:list_time.html.twig
                    datetime:
SonataAdminBundle:CRUD:list_datetime.html.twig
                    text:
SonataAdminBundle:CRUD:base_list_field.html.twig
                    trans:
SonataAdminBundle:CRUD:list_trans.html.twig
                    string:
SonataAdminBundle:CRUD:base_list_field.html.twig
                    smallint:
SonataAdminBundle:CRUD:base_list_field.html.twig
```


HCMC Office: 72/4 Truong Quoc Dung, District Phu Nhuan, HCMC, Viet Nam
☏: +84 8 39975901 🖷: +84 8 3997 5900
HK Office: Unit 706, 7/F., South Seas, Towers, 75 Mody Road, TsimShaTsui, Hong Kong
☏: +852 8197 1217
🖱: www.sutrixmedia.com

```
                    bigint:
SonataAdminBundle:CRUD:base_list_field.html.twig
                    integer:
SonataAdminBundle:CRUD:base_list_field.html.twig
                    decimal:
SonataAdminBundle:CRUD:base_list_field.html.twig
                    identifier:
SonataAdminBundle:CRUD:base_list_field.html.twig

                show:
                    array:
SonataAdminBundle:CRUD:show_array.html.twig
                    boolean:
SonataAdminBundle:CRUD:show_boolean.html.twig
                    date:
SonataAdminBundle:CRUD:show_date.html.twig
                    time:
SonataAdminBundle:CRUD:show_time.html.twig
                    datetime:
SonataAdminBundle:CRUD:show_datetime.html.twig
                    text:
SonataAdminBundle:CRUD:base_show_field.html.twig
                    trans:
SonataAdminBundle:CRUD:show_trans.html.twig
                    string:
SonataAdminBundle:CRUD:base_show_field.html.twig
                    smallint:
SonataAdminBundle:CRUD:base_show_field.html.twig
                    bigint:
SonataAdminBundle:CRUD:base_show_field.html.twig
                    integer:
SonataAdminBundle:CRUD:base_show_field.html.twig
                    decimal:
SonataAdminBundle:CRUD:base_show_field.html.twig
```

Config security:

```
# app/config/security.yml

jms_security_extra:
    secure_all_services: false
```


HCMC Office: 72/4 Truong Quoc Dung, District Phu Nhuan, HCMC, Viet Nam
☏: +84 8 39975901 🖷: +84 8 3997 5900
HK Office: Unit 706, 7/F., South Seas, Towers, 75 Mody Road, TsimShaTsui, Hong Kong
☏: +852 8197 1217
🖱: www.sutrixmedia.com

```
        expressions: true

    security:
        encoders:
            FOS\UserBundle\Model\UserInterface : sha512

        role_hierarchy:
            ROLE_ADMIN:       [ROLE_USER, ROLE_SONATA_ADMIN]
            ROLE_SUPER_ADMIN: [ROLE_ADMIN, ROLE_ALLOWED_TO_SWITCH]
            SONATA:
                - ROLE_SONATA_PAGE_ADMIN_PAGE_EDIT  # if you are
using acl then this line must be commented

        providers:
            in_memory:
                memory:
                    users:
                        user:  { password: userpass, roles: [
'ROLE_USER' ] }
                        admin: { password: adminpass, roles: [
'ROLE_ADMIN' ] }
            fos_userbundle:
                id: fos_user.user_manager

        firewalls:
            dev:
                pattern:  ^/(_(profiler|wdt)|css|images|js)/
                security: false

            login:
                pattern:  ^/demo/secured/login$
                security: false

            secured_area:
                pattern:    ^/demo/secured/
                form_login:
                    check_path: /demo/secured/login_check
                    login_path: /demo/secured/login
                logout:
                    path:   /demo/secured/logout
                    target: /demo/
```


HCMC Office: 72/4 Truong Quoc Dung, District Phu Nhuan, HCMC, Viet Nam
☏: +84 8 39975901 🖷: +84 8 3997 5900
HK Office: Unit 706, 7/F., South Seas, Towers, 75 Mody Road, TsimShaTsui, Hong Kong
☏: +852 8197 1217
🖱: www.sutrixmedia.com

```
                #anonymous: ~
                #http_basic:
                #    realm: "Secured Demo Area"

            # -> custom firewall for the admin area of the URL
            admin:
                switch_user:        true
                context:            user
                pattern:            /admin(.*)
                form_login:
                    provider:       fos_userbundle
                    login_path:     /admin/login
                    use_forward:    false
                    check_path:     /admin/login_check
                    failure_path:   null
                    use_referer:    true
                logout:
                    path:           /admin/logout
                    target:         /admin/login

                anonymous:    true
            # -> end custom configuration

            # defaut login area for standard users
            main:
                switch_user:        true
                context:            user
                pattern:            .*
                form_login:
                    provider:       fos_userbundle
                    login_path:     /login
                    use_forward:    false
                    check_path:     /login_check
                    failure_path:   null
                logout:             true
                anonymous:          true

        access_control:
            # URL of FOSUserBundle which need to be available to
anonymous users
            - { path: ^/_wdt, role: IS_AUTHENTICATED_ANONYMOUSLY }
```


HCMC Office: 72/4 Truong Quoc Dung, District Phu Nhuan, HCMC, Viet Nam
☏: +84 8 39975901 🖷: +84 8 3997 5900
HK Office: Unit 706, 7/F., South Seas, Towers, 75 Mody Road, TsimShaTsui, Hong Kong
☏: +852 8197 1217
🖱: www.sutrixmedia.com

```
            - { path: ^/_profiler, role: IS_AUTHENTICATED_ANONYMOUSLY
}
            - { path: ^/login$, role: IS_AUTHENTICATED_ANONYMOUSLY }

            # -> custom access control for the admin area of the URL
            - { path: ^/admin/login$, role:
IS_AUTHENTICATED_ANONYMOUSLY }
            - { path: ^/admin/logout$, role:
IS_AUTHENTICATED_ANONYMOUSLY }
            - { path: ^/admin/login-check$, role:
IS_AUTHENTICATED_ANONYMOUSLY }
            # -> end

            - { path: ^/register, role: IS_AUTHENTICATED_ANONYMOUSLY
}
            - { path: ^/resetting, role: IS_AUTHENTICATED_ANONYMOUSLY
}

            # Secured part of the site
            # This config requires being logged for the whole site
and having the admin role for the admin part.
            # Change these rules to adapt them to your needs
            - { path: ^/admin, role: [ROLE_ADMIN, ROLE_SONATA_ADMIN]
}
            - { path: ^/.*, role: IS_AUTHENTICATED_ANONYMOUSLY }

        acl:
            connection: default
```

Ok, now you can install web and clear cache to starting using:

```
    php app/console assets:install web
```

```
    php app/console cache:clear
```

Generate database to starting using Sonata admin:

```
    php app/console doctrine:schema:update --force
```


HCMC Office: 72/4 Truong Quoc Dung, District Phu Nhuan, HCMC, Viet Nam
☏: +84 8 39975901 🖷: +84 8 3997 5900
HK Office: Unit 706, 7/F., South Seas, Towers, 75 Mody Road, TsimShaTsui, Hong Kong
☏: +852 8197 1217
: www.sutrixmedia.com

Create supper admin for sonata admin (user: admin, password: password):

```
     php app/console fos:user:create admin admin@example.com
password –super-admin
```

Enable translator:

```
   # app/configs/config.yml
   framework:
       translator:      ~
```


HCMC Office: 72/4 Truong Quoc Dung,
District Phu Nhuan, HCMC, Viet Nam
☏: +84 8 39975901 🖷: +84 8 3997 5900
HK Office: Unit 706, 7/F., South Seas, Towers,
75 Mody Road, TsimShaTsui, Hong Kong
☏: +852 8197 1217
: www.sutrixmedia.com

7. Security & ACL

HTTP Authentication

The security component can be configured via your application configuration. In fact, most standard security setups are just a matter of using the right configuration. The following configuration tells Symfony to secure any URL matching /admin/* and to ask the user for credentials using basic HTTP authentication (i.e. the old-school username/password box):

The end result of this configuration is a fully-functional security system that looks like the following:

```
# app/config/security.yml
security:
    firewalls:
        secured_area:
            pattern:    ^/
            anonymous: ~
            http_basic:
                realm: "Secured Demo Area"

    access_control:
        - { path: ^/admin, roles: ROLE_ADMIN }

    providers:
        in_memory:
            memory:
                users:
                    ryan:  { password: ryanpass, roles: 'ROLE_USER' }
                    admin: { password: kitten, roles: 'ROLE_ADMIN' }

    encoders:
        Symfony\Component\Security\Core\User\User: plaintext
```

There are two users in the system (ryan and admin);

Users authenticate themselves via the basic HTTP authentication prompt;

Any URL matching /admin/* is secured, and only the admin user can access it;

All URLs *not* matching /admin/* are accessible by all users (and the user is never prompted to login).

Let's look briefly at how security works and how each part of the configuration comes into play.

Using a Traditional Login Form

So far, you've seen how to blanket your application beneath a firewall and then protect access to certain areas with roles. By using HTTP Authentication, you can effortlessly tap into the native

HCMC Office: 72/4 Truong Quoc Dung, District Phu Nhuan, HCMC, Viet Nam
☏: +84 8 39975901 🖷: +84 8 3997 5900
HK Office: Unit 706, 7/F., South Seas, Towers, 75 Mody Road, TsimShaTsui, Hong Kong
☏: +852 8197 1217
🖱: www.sutrixmedia.com

username/password box offered by all browsers. However, Symfony supports many authentication mechanisms out of the box.
In this section, you'll enhance this process by allowing the user to authenticate via a traditional HTML login form. First, enable form login under your firewall:

Now, when the security system initiates the authentication process, it will redirect the user to the login form
(

```
# app/config/security.yml
security:
    firewalls:
        secured_area:
            pattern:    ^/
            anonymous: ~
            form_login:
                login_path:  /login
                check_path:  /login_check
```

/login by default). Implementing this login form visually is your job. First, create two routes: one that will display the login form (i.e. /login) and one that will handle the login form submission (i.e. /login_check):

```
# app/config/routing.yml
login:
    pattern:   /login
    defaults:  { _controller: AcmeSecurityBundle:Security:login }
login_check:
    pattern:   /login_check
```

Securing Specific URL Patterns

The most basic way to secure part of your application is to secure an entire URL pattern. You've seen this already in the first example of this chapter, where anything matching the regular expression pattern ^/admin requires the ROLE_ADMIN role.

You can define as many URL patterns as you need - each is a regular expression.

For each incoming request, Symfony2 tries to find a matching access control rule (the first one wins). If the user isn't authenticated yet, the authentication process is initiated (i.e. the user is

HCMC Office: 72/4 Truong Quoc Dung, District Phu Nhuan, HCMC, Viet Nam
☏: +84 8 39975901 🖷: +84 8 3997 5900
HK Office: Unit 706, 7/F., South Seas, Towers, 75 Mody Road, TsimShaTsui, Hong Kong
☏: +852 8197 1217
🖱: www.sutrixmedia.com

given a chance to login). However, if the user

```
# app/config/security.yml
security:
    # ...
    access_control:
        - { path: ^/admin/users, roles: ROLE_SUPER_ADMIN }
        - { path: ^/admin, roles: ROLE_ADMIN }
```

is authenticated but doesn't have the required role, an AccessDeniedException exception is thrown, which you can handle and turn into a nice "access denied" error page for the user.

Since Symfony uses the first access control rule it matches, a URL like /admin/users/new will match the first rule and require only the ROLE_SUPER_ADMIN role. Any URL like /admin/blog will match the second rule and require ROLE_ADMIN.

Securing by IP

Certain situations may arise when you may need to restrict access to a given route based on IP. This is particularly relevant in the case of Edge Side Includes (ESI), for example, which utilize a route named "_internal". When ESI is used, the _internal route is required by the gateway cache to enable different caching options for subsections within a given page. This route comes with the ^/_internal prefix by default in the standard edition (assuming you've uncommented those lines from the routing file).
Here is an example of how you might secure this route from outside access:

```
# app/config/security.yml
security:
    # ...
    access_control:
        - { path: ^/_internal, roles: IS_AUTHENTICATED_ANONYMOUSLY, ip: 127.0.0.1 }
```

Securing by Channel

Much like securing based on IP, requiring the use of SSL is as simple as adding a new access_control entry:

Securing a Controller

```
# app/config/security.yml
security:
    # ...
    access_control:
        - { path: ^/cart/checkout, roles: IS_AUTHENTICATED_ANONYMOUSLY, requires_channel: https }
```


HCMC Office: 72/4 Truong Quoc Dung, District Phu Nhuan, HCMC, Viet Nam
☏: +84 8 39975901 🖷: +84 8 3997 5900
HK Office: Unit 706, 7/F., South Seas, Towers, 75 Mody Road, TsimShaTsui, Hong Kong
☏: +852 8197 1217
: www.sutrixmedia.com

Protecting your application based on URL patterns is easy, but may not be fine-grained enough in certain cases. When necessary, you can easily force authorization from inside a controller:

```
// ...
use Symfony\Component\Security\Core\Exception\AccessDeniedException;

public function helloAction($name)
{
    if (false === $this->get('security.context')->isGranted('ROLE_ADMIN')) {
        throw new AccessDeniedException();
    }

    // ...
}
```

You can also choose to install and use the optional JMSSecurityExtraBundle, which can secure your controller using annotations:

```
// ...
use JMS\SecurityExtraBundle\Annotation\Secure;

/**
 * @Secure(roles="ROLE_ADMIN")
 */
public function helloAction($name)
{
    // ...
}
```

Hierarchical Roles

Instead of associating many roles to users, you can define role inheritance rules by creating a role hierarchy:

```
# app/config/security.yml
security:
    role_hierarchy:
        ROLE_ADMIN:       ROLE_USER
        ROLE_SUPER_ADMIN: [ROLE_ADMIN, ROLE_ALLOWED_TO_SWITCH]
```

In the above configuration, users with ROLE_ADMIN role will also have the ROLE_USER role. The ROLE_SUPER_ADMIN role has ROLE_ADMIN, ROLE_ALLOWED_TO_SWITCH and ROLE_USER (inherited from ROLE_ADMIN).

Access Control in Templates

If you want to check if the current user has a role inside a template, use the built-in helper function:

HCMC Office: 72/4 Truong Quoc Dung, District Phu Nhuan, HCMC, Viet Nam
☏: +84 8 39975901 🖷: +84 8 3997 5900
HK Office: Unit 706, 7/F., South Seas, Towers, 75 Mody Road, TsimShaTsui, Hong Kong
☏: +852 8197 1217
🖱: www.sutrixmedia.com

```
{% if is_granted('ROLE_ADMIN') %}
    <a href="...">Delete</a>
{% endif %}
```

Access Control in Controllers

If you want to check if the current user has a role in your controller, use the isGranted() method of the security context:

```
public function indexAction()
{
    // show different content to admin users
    if ($this->get('security.context')->isGranted('ROLE_ADMIN')) {
        // ... load admin content here
    }

    // ... load other regular content here
}
```

ACL Sonata Admin configuration:

The security part is managed by a SecurityHandler, the bundle comes with 3 handlers

- sonata.admin.security.handler.role : ROLES to handle permissions
- sonata.admin.security.handler.acl : ACL and ROLES to handle permissions
- sonata.admin.security.handler.noop : always returns true, can be used with the Symfony2 firewall

We use ACL and ROLES to handle permission: # app/config/sonata.yml

```
security:
    handler: sonata.admin.security.handler.acl
    # role security information
    information:
        GUEST:    [LIST]
        STAFF:    [EDIT, LIST, CREATE]
        EDITOR:   [OPERATOR]
        ADMIN:    [MASTER]

    admin_permissions: [CREATE, LIST, DELETE, UNDELETE, OPERATOR, MASTER]
    object_permissions: [EDIT, DELETE, UNDELETE, OPERATOR, MASTER, OWNER]
```

information: # acl security information

admin_permissions:

object_permissions: # permission related to the objects

HCMC Office: 72/4 Truong Quoc Dung, District Phu Nhuan, HCMC, Viet Nam
☏: +84 8 39975901 🖷: +84 8 3997 5900
HK Office: Unit 706, 7/F., South Seas, Towers, 75 Mody Road, TsimShaTsui, Hong Kong
☏: +852 8197 1217
www.sutrixmedia.com

Setup ACL with the FOSUserBundle: Before you can use FriendsOfSymfony/FOSUserBundle you need to set it up as described in the documentation of the bundle.

```
namespace MS\Bundle\UserBundle\Entity;

use Doctrine\ORM\Mapping as ORM;
use Sonata\UserBundle\Entity\BaseUser;
use MS\Bundle\UserBundle\Security\Acl\Manager\AclManager;

/**
 * MS\Bundle\UserBundle\Entity\User
 *
 * @ORM\Table(name="ms_user")
 * @ORM\Entity
 */
class User extends BaseUser
{
    /**
     * @var integer $id
     *
     * @ORM\Id
     * @ORM\Column(type="integer")
     * @ORM\GeneratedValue(strategy="AUTO")
     */
    protected $id;
```

Note: Sonata\UserBundle\Entity\BaseUser extends FOS\UserBundle\Entity\User

In your app/config/config.yml you then need to put the following:

```
fos_user:
    db_driver:      orm # can be orm or odm
    firewall_name:  main
    user_class:     MS\Bundle\UserBundle\Entity\User
```

The following configuration for the SonataUserBundle defines:

the FriendsOfSymfony/FOSUserBundle as a security provider

the login form for authentification

the access control : resources with related required roles, the important part is the admin configuration

the acl option to enable the ACL.

HCMC Office: 72/4 Truong Quoc Dung, District Phu Nhuan, HCMC, Viet Nam
☏: +84 8 39975901 🖷: +84 8 3997 5900
HK Office: Unit 706, 7/F., South Seas, Towers, 75 Mody Road, TsimShaTsui, Hong Kong
☏: +852 8197 1217
🖱: www.sutrixmedia.com

the AdminPermissionMap defines the permissions of the Admin class

In your app/config/sonata.yml you then need to put the following:

```
parameters:
    sonata.user.admin.user.entity: MS\Bundle\UserBundle\Entity\User
    sonata.user.admin.group.entity: MS\Bundle\UserBundle\Entity\Group
    # optionally use a custom MaskBuilder
    sonata.admin.security.mask.builder.class: Sonata\AdminBundle\Security\Acl\Permission\MaskBuilder
    security.acl.permission.map.class: Sonata\AdminBundle\Security\Acl\Permission\AdminPermissionMap
```

In app/config/security.yml:

```
security:
    encoders:
        FOS\UserBundle\Model\UserInterface : sha512

    role_hierarchy:
        ROLE_ADMIN:       [ROLE_USER, ROLE_SONATA_ADMIN]
        ROLE_SUPER_ADMIN: [ROLE_ADMIN, ROLE_ALLOWED_TO_SWITCH]
        #database role
        ROLE_EDITOR_ADMIN: [ROLE_ADMIN, ROLE_MS_NEWS_ARTICLE_ADMIN]
        ROLE_MANAGER_ADMIN: [ROLE_ADMIN, ROLE_EDITOR_ADMIN]
        SONATA:
            - ROLE_SONATA_PAGE_ADMIN_PAGE_EDIT

    providers:
        in_memory:
            memory:
                users:
                    user:  { password: userpass, roles: [ 'ROLE_USER' ] }
                    admin: { password: adminpass, roles: [ 'ROLE_ADMIN' ] }
        fos_userbundle:
            id: fos_user.user_manager
```


HCMC Office: 72/4 Truong Quoc Dung, District Phu Nhuan, HCMC, Viet Nam
☏: +84 8 39975901 🖷: +84 8 3997 5900
HK Office: Unit 706, 7/F., South Seas, Towers, 75 Mody Road, TsimShaTsui, Hong Kong
☏: +852 8197 1217
🖱: www.sutrixmedia.com

```
firewalls:
    admin:
        switch_user:        true
        context:            user
        pattern:            /admin(.*)
        form_login:
            provider:       fos_userbundle
            login_path:     /admin/login
            use_forward:    false
            check_path:     /admin/login_check
            failure_path:   null
            use_referer:    true
        logout:
            path:           /admin/logout
            target:         /admin/login

        anonymous:    true
    # -> end custom configuration

    # defaut login area for standard users
    main:
        switch_user:        true
        context:            user
        pattern:            .*
        form_login:
            provider:       fos_userbundle
            login_path:     /login
            use_forward:    false
            check_path:     /login_check
            failure_path:   null
        logout:             true
        anonymous:          true
```


HCMC Office: 72/4 Truong Quoc Dung, District Phu Nhuan, HCMC, Viet Nam
☏: +84 8 39975901 🖷: +84 8 3997 5900
HK Office: Unit 706, 7/F., South Seas, Towers, 75 Mody Road, TsimShaTsui, Hong Kong
☏: +852 8197 1217
🖱: **www.sutrixmedia.com**

```
access_control:
    # URL of FOSUserBundle which need to be available to anonymous users
    - { path: ^/_wdt, role: IS_AUTHENTICATED_ANONYMOUSLY }
    - { path: ^/_profiler, role: IS_AUTHENTICATED_ANONYMOUSLY }
    - { path: ^/login$, role: IS_AUTHENTICATED_ANONYMOUSLY }

    # -> custom access control for the admin area of the URL
    - { path: ^/admin/login$, role: IS_AUTHENTICATED_ANONYMOUSLY }
    - { path: ^/admin/logout$, role: IS_AUTHENTICATED_ANONYMOUSLY }
    - { path: ^/admin/login-check$, role: IS_AUTHENTICATED_ANONYMOUSLY }
    # -> end

    - { path: ^/register, role: IS_AUTHENTICATED_ANONYMOUSLY }
    - { path: ^/resetting, role: IS_AUTHENTICATED_ANONYMOUSLY }

    # Secured part of the site
    # This config requires being logged for the whole site and having
    # the admin role for the admin part.
    # Change these rules to adapt them to your needs
    - { path: ^/admin, role: [ROLE_ADMIN, ROLE_SONATA_ADMIN] }
    - { path: ^/.*, role: IS_AUTHENTICATED_ANONYMOUSLY }

acl:
    connection: default
```

Install the ACL tables: $ php app/console sonata:admin:setup-acl

```
Starting ACL AdminBundle configuration
 > install ACL for sonata.user.admin.user
   - update role: ROLE_SONATA_USER_ADMIN_USER_GUEST, permissions: ["LIST"]
   - update role: ROLE_SONATA_USER_ADMIN_USER_STAFF, permissions: ["LIST","CREATE"]
   - update role: ROLE_SONATA_USER_ADMIN_USER_EDITOR, permissions: ["OPERATOR"]
   - update role: ROLE_SONATA_USER_ADMIN_USER_ADMIN, permissions: ["MASTER"]
 > install ACL for sonata.user.admin.group
   - update role: ROLE_SONATA_USER_ADMIN_GROUP_GUEST, permissions: ["LIST"]
   - update role: ROLE_SONATA_USER_ADMIN_GROUP_STAFF, permissions: ["LIST","CREATE"]
   - update role: ROLE_SONATA_USER_ADMIN_GROUP_EDITOR, permissions: ["OPERATOR"]
   - update role: ROLE_SONATA_USER_ADMIN_GROUP_ADMIN, permissions: ["MASTER"]
 > install ACL for sonata.user.admin.role
   - update role: ROLE_SONATA_USER_ADMIN_ROLE_GUEST, permissions: ["LIST"]
   - update role: ROLE_SONATA_USER_ADMIN_ROLE_STAFF, permissions: ["LIST","CREATE"]
   - update role: ROLE_SONATA_USER_ADMIN_ROLE_EDITOR, permissions: ["OPERATOR"]
   - update role: ROLE_SONATA_USER_ADMIN_ROLE_ADMIN, permissions: ["MASTER"]
```

Create a new root user and groups: $ php app/console msuser:init

```
update group Administrators
update group Manager
update group Editor
update user: username=admin, password=password
```

If you already have objects, you can generate the object ACL rules for each object of an admin: $ php app/console sonata:admin:generate-object-acl

HCMC Office: 72/4 Truong Quoc Dung, District Phu Nhuan, HCMC, Viet Nam
☏: +84 8 39975901 🖷: +84 8 3997 5900
HK Office: Unit 706, 7/F., South Seas, Towers, 75 Mody Road, TsimShaTsui, Hong Kong
☏: +852 8197 1217
🖱: www.sutrixmedia.com

```
> generate ACLs for sonata.user.admin.user
  - [TOTAL] generated class ACEs for 1 objects (added 0, updated 1)
> generate ACLs for sonata.user.admin.group
  - [TOTAL] generated class ACEs for 3 objects (added 3, updated 0)
> generate ACLs for sonata.user.admin.role
  - [TOTAL] generated class ACEs for 3 objects (added 3, updated 0)
> generate ACLs for ms.news.category
  - [TOTAL] generated class ACEs for 1 objects (added 1, updated 0)
> generate ACLs for ms.news.article
  - [TOTAL] generated class ACEs for 0 objects (added 0, updated 0)
```

Roles and Access control lists

A user can have several roles when working with an application. Each Admin class has several roles, and each role specifies the permissions of the user for the Admin class. Or more specifically, what the user can do with the domain object(s) the Admin class is created for.

By default each Admin class contains the following roles, override the property $securityInformation to change this:

ROLE_SONATA_..._GUEST : a guest that is allowed to view an object and a list of objects;

ROLE_SONATA_..._STAFF : probably the biggest part of the users, a staff user has the same permissions as guests and is additionally allowed to EDIT and CREATE new objects;

ROLE_SONATA_..._EDITOR: an editor is granted all access and, compared to the staff users, is allowed to DELETE;

- ROLE_SONATA_..._ADMIN : an administrative user is granted all access and on top of that, the user is allowed to grant other users access.

Usage

Everytime you create a new Admin class, you should start with the command php app/console sonata:admin:setup-acl so the ACL database will be updated with the latest roles and permissions.

In the templates, or in your code, you can use the Admin method isGranted() :

check for an admin that the user is allowed to EDIT :

```
{# use the admin security method  #}
{% if admin.isGranted('EDIT') %} {# ... #} {% endif %}

{# or use the default is_granted symfony helper, the following will give the same result #}
{% if is_granted('ROLE_SUPER_ADMIN') or is_granted('EDIT', admin) %} {# ... #} {% endif %}
```


HCMC Office: 72/4 Truong Quoc Dung, District Phu Nhuan, HCMC, Viet Nam
☏: +84 8 39975901 🖷: +84 8 3997 5900
HK Office: Unit 706, 7/F., South Seas, Towers, 75 Mody Road, TsimShaTsui, Hong Kong
☏: +852 8197 1217
🖱: www.sutrixmedia.com

check for an admin that the user is allowed to DELETE, the object is added to also check if the object owner is allowed to DELETE :

```
{# use the admin security method  #}
{% if admin.isGranted('DELETE', object) %} {# ... #} {% endif %}

{# or use the default is_granted symfony helper, the following will give the same result #}
{% if is_granted('ROLE_SUPER_ADMIN') or is_granted('DELETE', object) %} {# ... #} {% endif %}
```

ACL on Frontend

Now, before you can finally get into action, you need to do some bootstrapping. First, you need to configure the connection the ACL system is supposed to use: # in app/config/sonata.yml or in app/config/config.yml

```
security:
    acl:
        connection: default
```

Creating an ACL and adding an ACE

HCMC Office: 72/4 Truong Quoc Dung,
District Phu Nhuan, HCMC, Viet Nam
☏: +84 8 39975901 🖷: +84 8 3997 5900
HK Office: Unit 706, 7/F., South Seas, Towers,
75 Mody Road, TsimShaTsui, Hong Kong
☏: +852 8197 1217
🖱: www.sutrixmedia.com

```
use Symfony\Bundle\FrameworkBundle\Controller\Controller;
use Symfony\Component\Security\Core\Exception\AccessDeniedException;
use Symfony\Component\Security\Acl\Domain\ObjectIdentity;
use Symfony\Component\Security\Acl\Domain\UserSecurityIdentity;
use Symfony\Component\Security\Acl\Permission\MaskBuilder;

class BlogController
{
    // ...

    public function addCommentAction(Post $post)
    {
        $comment = new Comment();

        // ... setup $form, and bind data

        if ($form->isValid()) {
            $entityManager = $this->getDoctrine()->getManager();
            $entityManager->persist($comment);
            $entityManager->flush();

            // creating the ACL
            $aclProvider = $this->get('security.acl.provider');
            $objectIdentity = ObjectIdentity::fromDomainObject($comment);
            $acl = $aclProvider->createAcl($objectIdentity);

            // retrieving the security identity of the currently logged-in user
            $securityContext = $this->get('security.context');
            $user = $securityContext->getToken()->getUser();
            $securityIdentity = UserSecurityIdentity::fromAccount($user);

            // grant owner access
            $acl->insertObjectAce($securityIdentity, MaskBuilder::MASK_OWNER);
            $aclProvider->updateAcl($acl);
        }
    }
}
```

There are a couple of important implementation decisions in this code snippet. For now, I only want to highlight two:

First, you may have noticed that ->createAcl() does not accept domain objects directly, but only implementations of the ObjectIdentityInterface. This additional step of indirection allows you to work with ACLs even when you have no actual domain object instance at hand. This will be extremely helpful if you want to check permissions for a large number of objects without actually hydrating these objects.

The other interesting part is the ->insertObjectAce() call. In the example, you are granting the user who is currently logged in owner access to the Comment. The MaskBuilder::MASK_OWNER is a pre-defined integer bitmask; don't worry the mask builder will abstract away most of the

HCMC Office: 72/4 Truong Quoc Dung, District Phu Nhuan, HCMC, Viet Nam
☏: +84 8 39975901 🖷: +84 8 3997 5900
HK Office: Unit 706, 7/F., South Seas, Towers, 75 Mody Road, TsimShaTsui, Hong Kong
☏: +852 8197 1217
🖱: www.sutrixmedia.com

technical details, but using this technique you can store many different permissions in one database row which gives a considerable boost in performance.

Checking Access

In this example, you check whether the user has the

```
// ...

class BlogController
{
    // ...

    public function editCommentAction(Comment $comment)
    {
        $securityContext = $this->get('security.context');

        // check for edit access
        if (false === $securityContext->isGranted('EDIT', $comment))
        {
            throw new AccessDeniedException();
        }

        // ... retrieve actual comment object, and do your editing here
    }
}
```

EDIT permission. Internally, Symfony2 maps the permission to several integer bitmasks, and checks whether the user has any of them

Cumulative Permissions

In the first example above, you only granted the user the OWNER base permission. While this effectively also allows the user to perform any operation such as view, edit, etc. on the domain object, there are cases where you may want to grant these permissions explicitly.

The MaskBuilder can be used for creating bit masks easily by combining several base permissions:

```
$builder = new MaskBuilder();
$builder
    ->add('view')
    ->add('edit')
    ->add('delete')
    ->add('undelete')
;
$mask = $builder->get(); // int(29)
```

This integer bitmask can then be used to grant a user the base permissions you added above:

HCMC Office: 72/4 Truong Quoc Dung, District Phu Nhuan, HCMC, Viet Nam
☏: +84 8 39975901 🖷: +84 8 3997 5900
HK Office: Unit 706, 7/F., South Seas, Towers, 75 Mody Road, TsimShaTsui, Hong Kong
☏: +852 8197 1217
: www.sutrixmedia.com

The user is now allowed to view, edit, delete, and un-delete objects.

```
$identity = new UserSecurityIdentity('johannes', 'Acme\UserBundle\Entity\User');
$acl->insertObjectAce($identity, $mask);
```


HCMC Office: 72/4 Truong Quoc Dung,
District Phu Nhuan, HCMC, Viet Nam
☏: +84 8 39975901 🖷: +84 8 3997 5900
HK Office: Unit 706, 7/F., South Seas, Towers,
75 Mody Road, TsimShaTsui, Hong Kong
☏: +852 8197 1217
🖱: **www.sutrixmedia.com**

8. Twig Extension

a) Step 1: Create the Extension Class:

To get your custom functionality you must first create a Twig Extension class. As an example you'll create a price filter to format a given number into price:

```
<?php
# src/MS/Bundle/NewsBundle/Twig/MSTwigNewsExtension.php

namespace MS\Bundle\NewsBundle\Twig;


class MSTwigNewsExtension extends \Twig_Extension
{
    public function getName()
    {
        return 'ms.twig.news_extension'; //
    }

    public function getFunctions()
    {
        return array(
            'substr' => new \Twig_Function_Method($this,
'subString'),
        );
    }

    public function getFilters()
    {
        return array(
            'cutword' => new \Twig_Filter_Method($this,
'cutWordString'),
        );
    }

    public function subString($str, $start = 0, $length = null)
    {
```


HCMC Office: 72/4 Truong Quoc Dung, District Phu Nhuan, HCMC, Viet Nam
☏: +84 8 39975901 🖷: +84 8 3997 5900
HK Office: Unit 706, 7/F., South Seas, Towers, 75 Mody Road, TsimShaTsui, Hong Kong
☏: +852 8197 1217
: www.sutrixmedia.com

```
            if(null === $length){
                return substr($str, $start);
            } else {
                return substr($str, $start, $length);
            }
        }

        function cutWordString($str, $length, $append = ' …',
$breakWords = TRUE)
        {
            $strLength = mb_strlen($str);

            if ($strLength <= $length) {
                return $str;
            }

            if ( $breakWords) {
                while ($length < $strLength AND preg_match('/^\pL$/',
mb_substr($str, $length, 1))) {
                    $length++;
                }
            }

            return mb_substr($str, 0, $length) . $append;
        }
    }
```

We have two functions above: getFunctions() and getFilters().

b) Step 2: Register an Extension as a Service

```
# src/MS/Bundle/NewsBundle/Resources/config/services.yml
services:
    ms.twig.news_extension:
        class: MS\Bundle\NewsBundle\Twig\MSTwigNewsExtension
        tags:
            - { name: twig.extension }
```

c) Step 3: Using the custom Extension

{{ foo.content | cutword(50) }}

HCMC Office: 72/4 Truong Quoc Dung, District Phu Nhuan, HCMC, Viet Nam
☏: +84 8 39975901 🖷: +84 8 3997 5900
HK Office: Unit 706, 7/F., South Seas, Towers, 75 Mody Road, TsimShaTsui, Hong Kong
☏: +852 8197 1217
🖱: www.sutrixmedia.com

```
{{  substr(foo.content, 2, 20) }}
```


HCMC Office: 72/4 Truong Quoc Dung,
District Phu Nhuan, HCMC, Viet Nam
☏: +84 8 39975901 🖷: +84 8 3997 5900
HK Office: Unit 706, 7/F., South Seas, Towers,
75 Mody Road, TsimShaTsui, Hong Kong
☏: +852 8197 1217
🖱: **www.sutrixmedia.com**