# **How to install and configure Symfony 2.1 + Sonata Admin + FosUser**

**Change Tracking:**

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Author | Note |
| 07/12/2012 | 0.1 | Duy Nguyen | Create base document. |
|  |  |  |  |

Before setup, you need install **git version** and can **run git anywhere in command line**. Same for php

1. **Install Symfony 2.1**

**We have 2 ways to download and install symfony:**

* 1. 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:

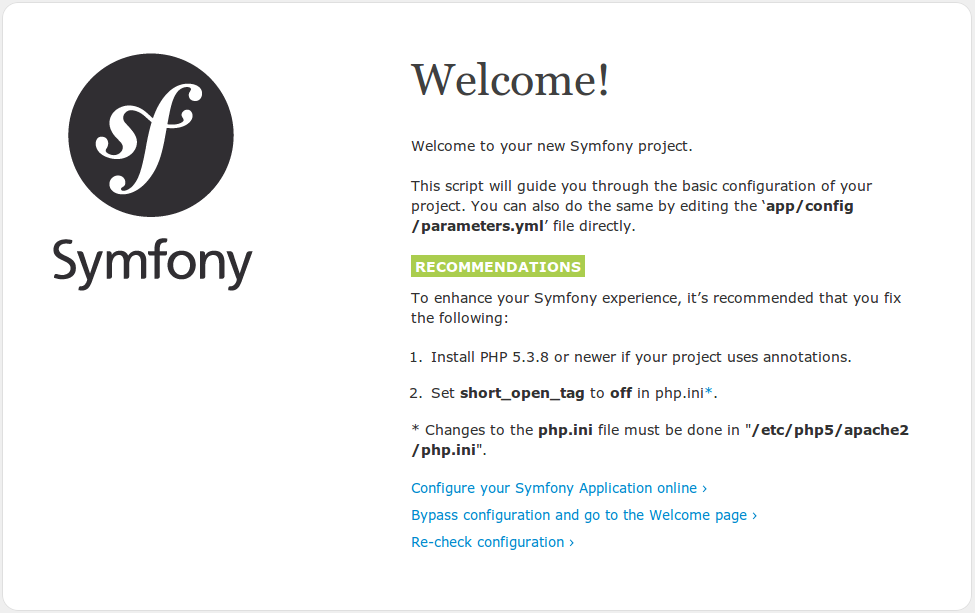
php composer.phar create-project symfony/framework-standard-edition symfony 2.1.4

* 1. 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



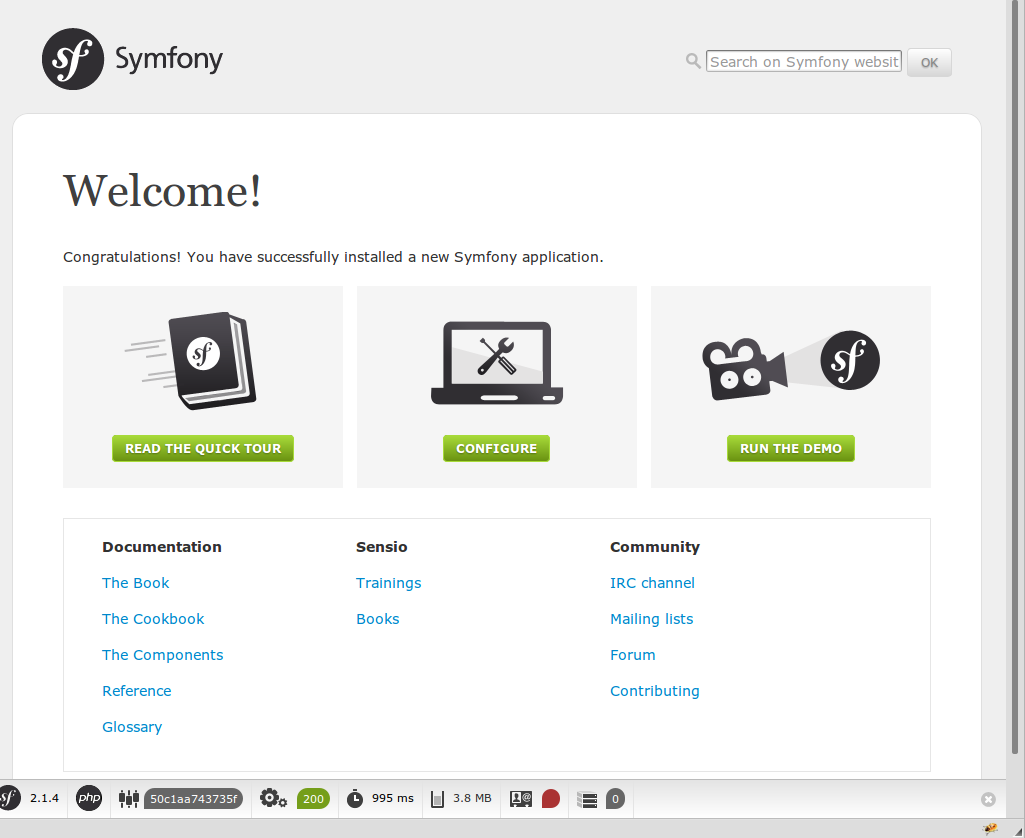
This page show for you how to enhance your Symfony experience, it’s recommended that you fix your server.

**Config symfony work with database:**

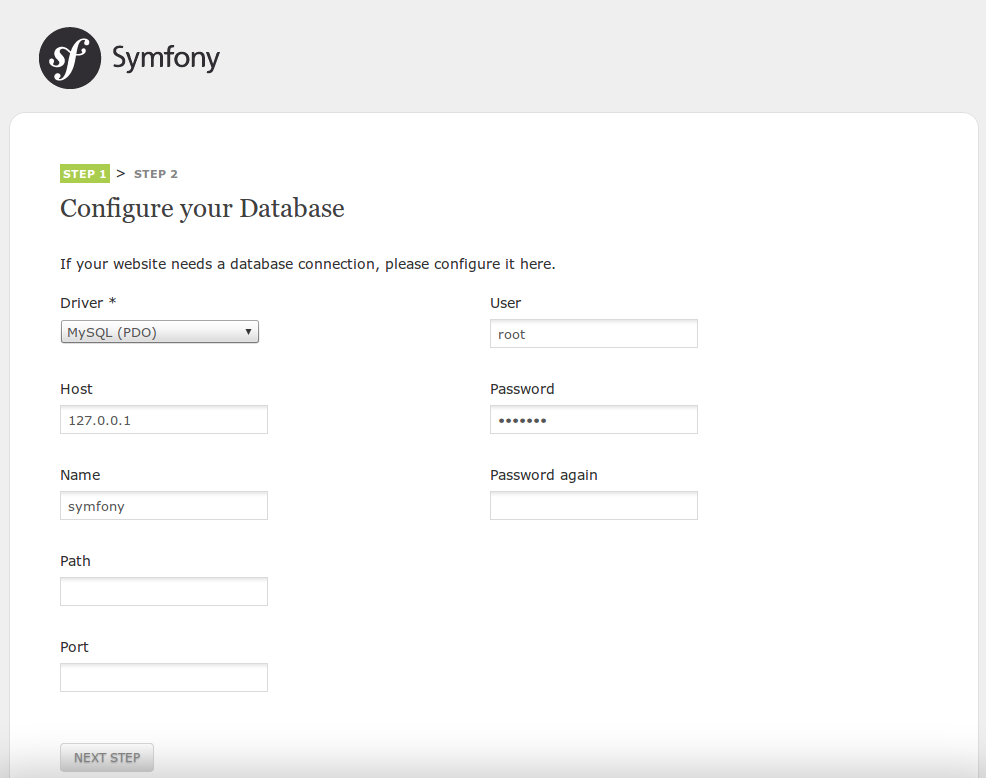
* + When you completed download symfony, go to:

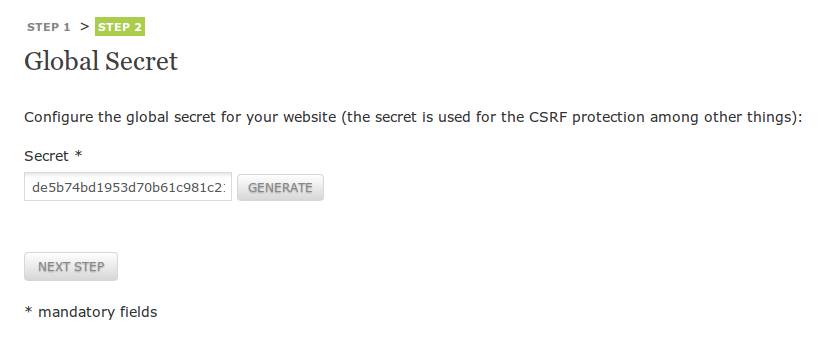
http://localhost/symfony\_path/web/app\_dev.php/

* + Click to configure to config symfony in your server:

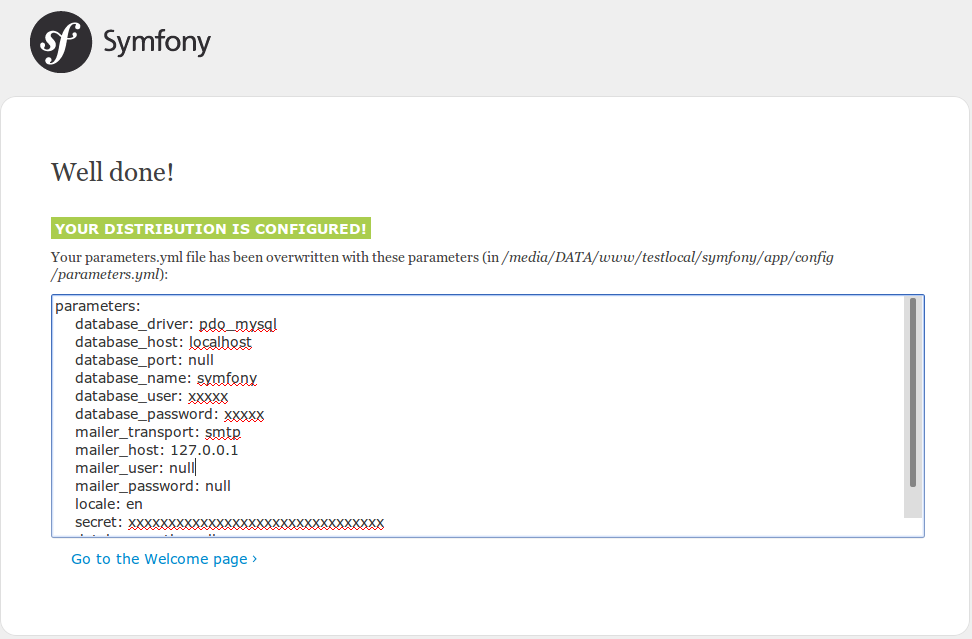


Follow with screen to config your database. Click next step to set up your Global Secret





If done you will see



1. **Install Sonata Admin + FosUser**
   1. **Install Sonata Admin**
      * 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.
      * In 1.1 download composer.phar and copy it to folder of symfony (same level with folder app, web, …)
      * 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:

A description...

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

"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",**

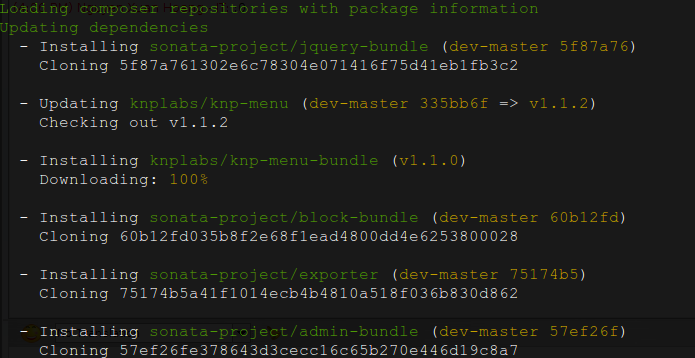
**"sonata-project/easy-extends-bundle": "dev-master",**

**"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:



* 1. **Config Sonata admin**

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:

# 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

If you see a error

[Symfony\Component\Config\Definition\Exception\InvalidConfigurationException]

The child node "db\_driver" at path "fos\_user" must be configured.

Please copy code below to paste to file app/config.yml

fos\_user:

db\_driver: orm # can be orm or odm

firewall\_name: main

user\_class: MS\Bundle\UserBundle\Entity\User

After that, run the command again

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;

/\*\*

\* 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;

/\*\*

\* @var groups

\*

\* @ORM\ManyToMany(targetEntity="Group")

\* @ORM\JoinTable(name="ms\_users\_groups")

\*/

protected $groups;

/\*\*

\* Get id

\*

\* @return integer

\*/

public function getId()

{

return $this->id;

}

}

Create the entity Group

# src\ MS\Bundle\UserBundle\Entity\Group.php

<?php

namespace MS\Bundle\UserBundle\Entity;

use Sonata\UserBundle\Entity\BaseGroup as BaseGroup;

use Doctrine\ORM\Mapping as ORM;

use Doctrine\Common\Collections\ArrayCollection;

/\*\*

\* MS\Bundle\UserBundle\Entity\Group

\*

\* @ORM\Entity

\* @ORM\Table(name="ms\_group")

\*/

class Group extends BaseGroup

{

/\*\*

\* @var integer $id

\*

\* @ORM\Column(name="id", type="integer")

\* @ORM\Id

\* @ORM\GeneratedValue(strategy="AUTO")

\*/

protected $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

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:

# 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

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

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/

#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 }

- { 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

Please run above command that to create database.

php app/console doctrine:database:create

Please run above command that to create table

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

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: ~

Now you can go to url to login

<http://localhost/symfony_path/app_dev.php/admin/login>

When you click on menu user/list If you see a error

Please adding the code

sonata.user.admin.role.translation\_domain: MSUserBundle

into file MS/Bundle/UserBundle/Resources/config/services.yml