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:
   1.1. Install symfony with composer.phar:

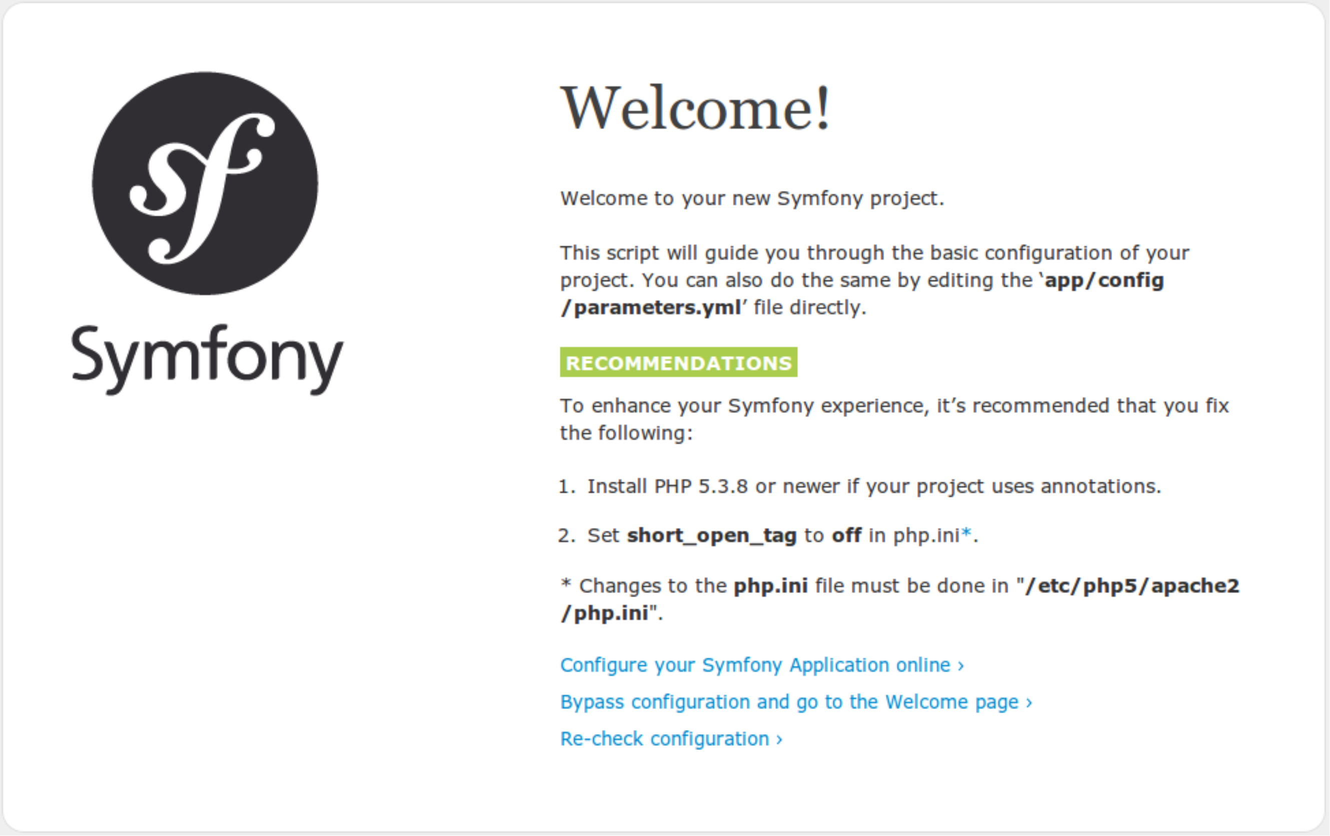
   – Go to http://getcomposer.org/download/ to download composer.phar. Copy it to folder you want install and run command line at that folder:

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

   1.2. 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
```



Welcome!

Welcome to your new Symfony project.

This script will guide you through the basic configuration of your project. You can also do the same by editing the '**app/config/parameters.yml**' file directly.

RECOMMENDATIONS

To enhance your Symfony experience, it's recommended that you fix the following:

1. Install PHP 5.3.8 or newer if your project uses annotations.
2. Set **short_open_tag** to **off** in php.ini*.

* Changes to the **php.ini** file must be done in "**/etc/php5/apache2/php.ini**".

Configure your Symfony Application online ›
Bypass configuration and go to the Welcome page ›
Re-check configuration ›

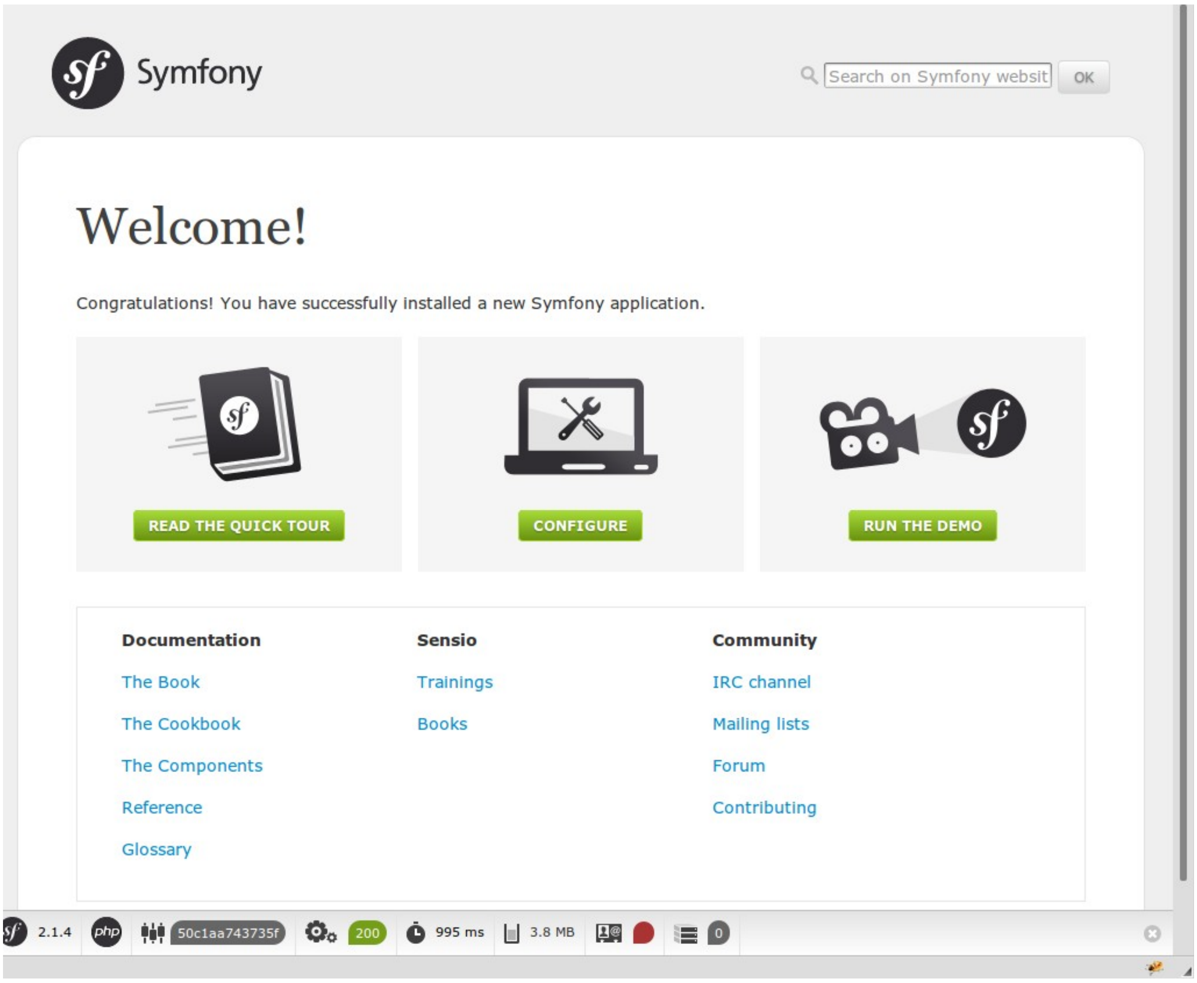
This page show for you how to enhance your Symfony experience, it's recommended that you fix your server.

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:



Welcome!

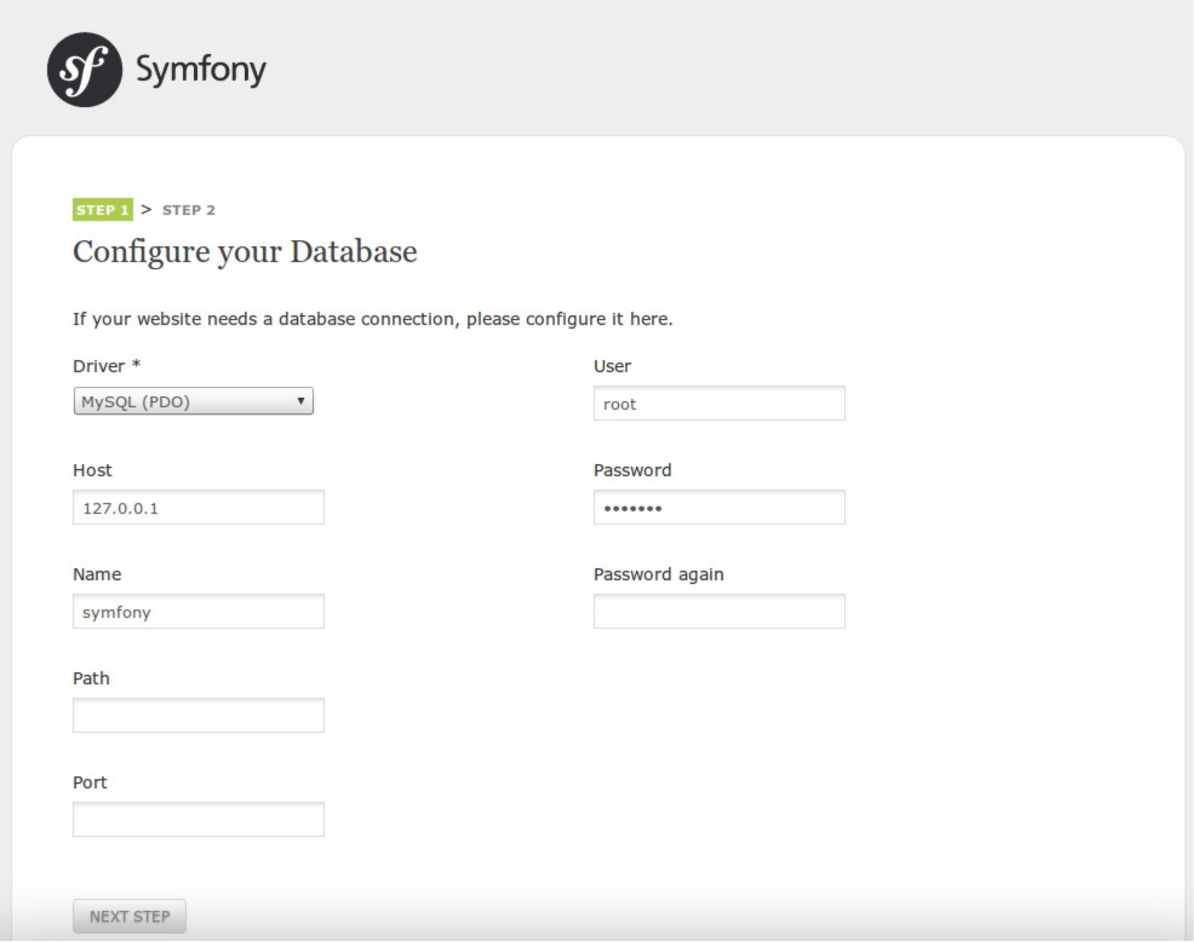
Congratulations! You have successfully installed a new Symfony application.

Documentation
- The Book
- The Cookbook
- The Components
- Reference
- Glossary

Sensio
- Trainings
- Books

Community
- IRC channel
- Mailing lists
- Forum
- Contributing

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



Configure your Database

If your website needs a database connection, please configure it here.

Driver * MySQL (PDO)

User root

Host 127.0.0.1

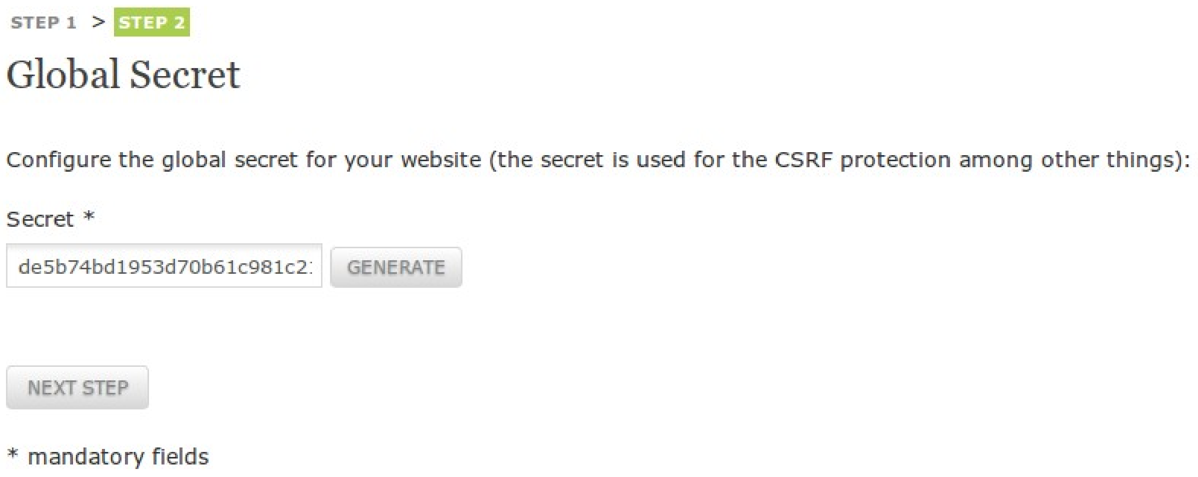
Password •••••••

Name symfony

Password again

Path

Port



Global Secret

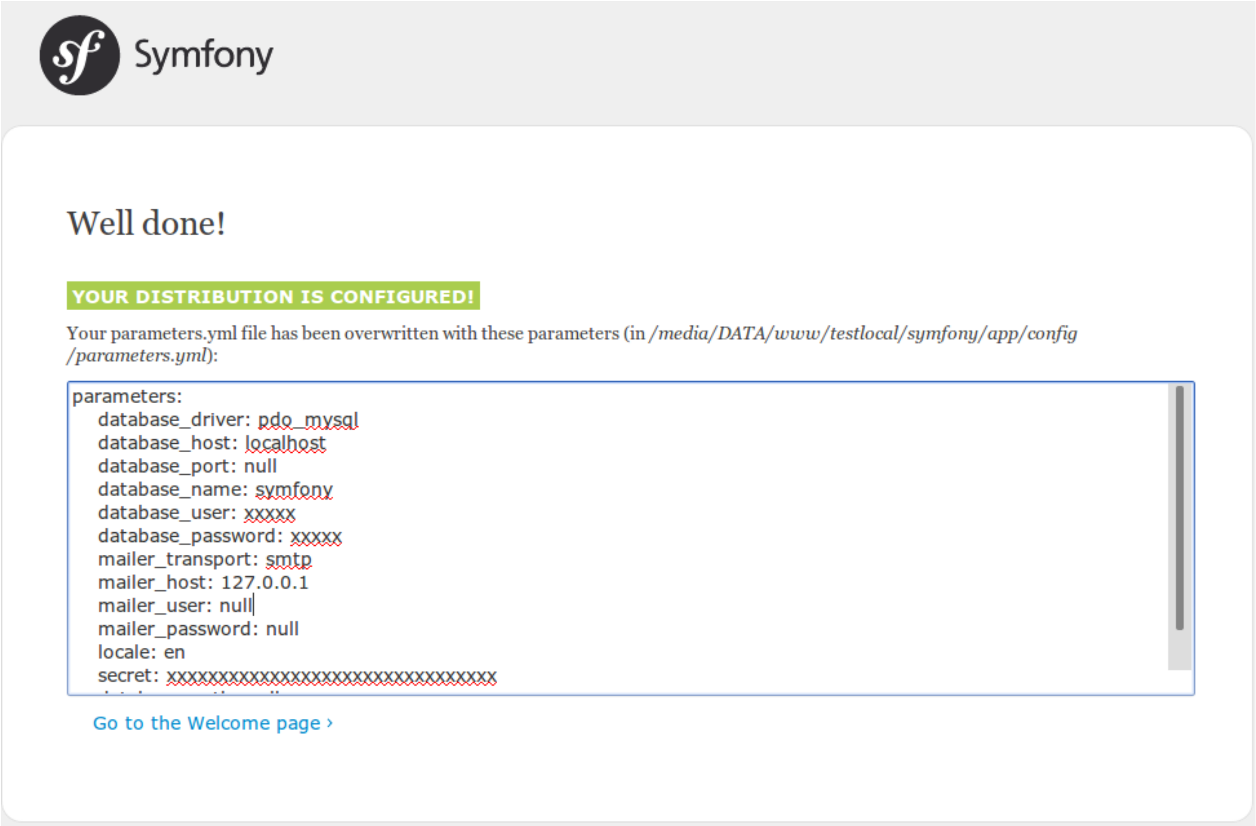
Configure the global secret for your website (the secret is used for the CSRF protection among other things):

Secret *

de5b74bd1953d70b61c981c2

* mandatory fields

If done you will see



Well done!

YOUR DISTRIBUTION IS CONFIGURED!

Your parameters.yml file has been overwritten with these parameters (in */media/DATA/www/testlocal/symfony/app/config/parameters.yml*):

```
parameters:
    database_driver: pdo_mysql
    database_host: localhost
    database_port: null
    database_name: symfony
    database_user: xxxxx
    database_password: xxxxx
    mailer_transport: smtp
    mailer_host: 127.0.0.1
    mailer_user: null
    mailer_password: null
    locale: en
    secret: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

2. Install Sonata Admin + FosUser

2.1. 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
```

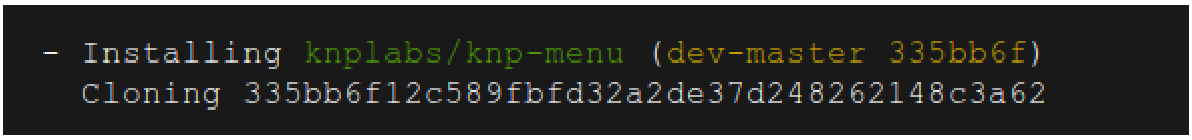
In folder of symfony, open composer.json add:

```
    "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:



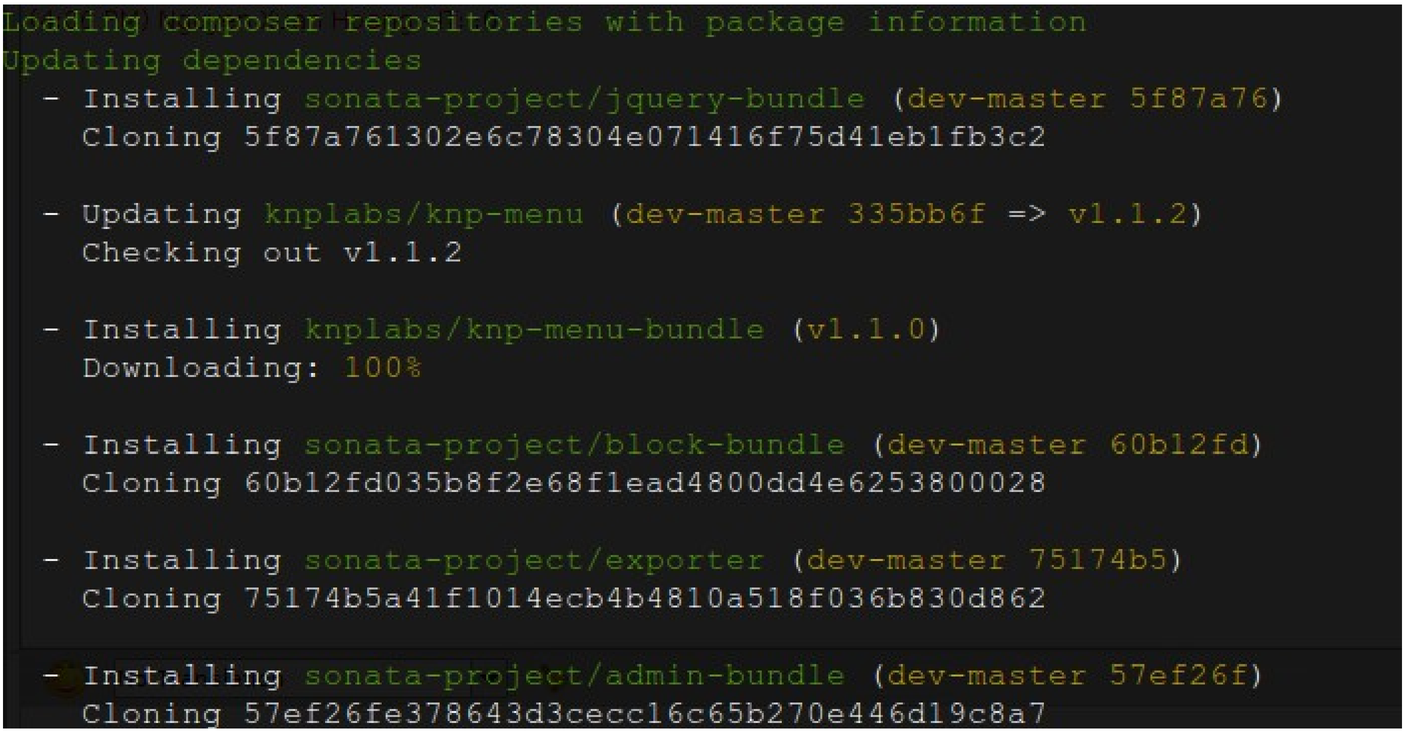
```
- Installing knplabs/knp-menu (dev-master 335bb6f)
  Cloning 335bb6f12c589fbfd32a2de37d248262148c3a62
```

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:



```
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
```

2.2. 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
```

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;


    /**
     * 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
```

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
```

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
```