— tech du peupl

Language : English Français

By: victor 04 July 2013 Symfony2.3 Share

[2/4] Symfony 2.3, SonataAdminBundle, SonataMediabundle, SonataUserBundle and FOSUserBundle : a good project base

Part 2 of 4.

In this post, we will start our project by installing all we need.
Let's see the summary again, for those trying to catch up :

1. Symfony 2.3 : ACL, SonataAdminBundle, SonataMediabundle, SonataUserBundle and FOSUserBundle
2. **Symfony 2.3, SonataAdminBundle, SonataMediabundle, SonataUserBundle and FOSUserBundle : a good project base**
3. Symfony 2.3 : Customizing SonataAdminBundle
4. Symfony 2.3, SonataAdminBundle, FOSUserBundle and ACLs

Create a new Symfony 2.3 project named "tech".

```
♀ composer.phar create-project symfony/framework-standard-edition
./tech 2.3.0
♀ cd tech/
```

Set read-write-execute permissions to app/cache and app/logs for both your current user and http (which is my Apache user). (Some of you, e.g. ubuntu users, should put www-data instead of http in these two commands.)

```
♀ su -c "setfacl -R -m u:http:rwx -m u:whoami:rwx app/cache app/logs"
♀ su -c "setfacl -dR -m u:http:rwx -m u:whoami:rwx app/cache app/logs"
```

Create our TechNewsBundle :

Search

Language

English
Français

Recent entries

[4/4] Symfony 2.3, SonataAdminBundle, FOSUserBundle and ACLs 08-07-2013
[3/4] Symfony 2.3 : Customizing SonataAdminBundle 04-07-2013
[2/4] Symfony 2.3, SonataAdminBundle, SonataMediabundle, SonataUserBundle and FOSUserBundle : a good project base 04-07-2013
[1/4] Symfony 2.3 : ACL, SonataAdminBundle, SonataMediabundle, SonataUserBundle and FOSUserBundle 04-07-2013
Symfony 2.1 : Customizing SonataAdminBundle 03-12-2012

Recent comments

sagitta on [2/4] Symfony 2.3, SonataAdminBundle, SonataMediabundle, SonataUserBundle and FOSUserBundle : a good project base 13-08-2013
James on [2/4] Symfony 2.3, SonataAdminBundle, SonataMediabundle, SonataUserBundle and FOSUserBundle : a good project base 09-08-2013
Bob Door on [4/4] Symfony 2.3, SonataAdminBundle, FOSUserBundle and ACLs 09-08-2013
Guest on [4/4] Symfony 2.3, SonataAdminBundle, FOSUserBundle and ACLs 03-08-2013
no name on [4/4] Symfony 2.3, SonataAdminBundle, FOSUserBundle and ACLs 17-07-2013

```
⚘ php app/console generate:bundle
Bundle namespace: Tech/NewsBundle
```

Accept all default choices except :

```
Configuration format (yml, xml, php, or annotation): annotation
Do you want to generate the whole directory structure [no]? yes
```

Then we need an News entity :

```
⚘ php app/console generate:doctrine:entity
The Entity shortcut name: TechNewsBundle:News
```

Add a field "*title*", accept the default "*string*" and "*255*" and that's it.

Time to install some fancy bundles : FOSUserBundle, SonataUserBundle, SonataAdminBundle and SonataMediaBundle. This step can be somewhat tricky for newcomers, because you cannot install everything right away. You have to get some bundles first, configure them, install another one, configure it, generate something, move it somewhere, install the next ones, and so on. Do it as follows or do it as each bundle's doc says, but don't rush and don't skip steps.

a. FOSUserBundle

→ Official documentation

In *config.yml*, uncomment `translator: { fallback: "%locale%" }`

composer.json : add `"friendsofsymfony/user-bundle": "1.3.*",` to require section.
Run *composer update*. Register FOSUB in *AppKernel.php* : `new FOS\UserBundle\FOSUserBundle(),`

In order to use FOSUserBundle, our bundle needs a User entity. In a real project, you would probably want to create your own UserBundle containing this user entity and your users' logic, so that you could use these same users later on when implementing news comments, etc. We will instead create this user entity in our NewsBundle for the sake of simplicity.

Create the following file :

```
// src/Tech/NewsBundle/Entity/User.php

namespace Tech\NewsBundle\Entity;

use FOS\UserBundle\Entity\User as BaseUser;
use Doctrine\ORM\Mapping as ORM;

```

↣ Other blogs you might like !

Categories

CSS
JS
Symfony2.1
Symfony2.3
Tools

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

    public function __construct()
    {
        parent::__construct();
        // your own logic
    }
}
```

Replace the security section by this one :

```
#security.yml
security:
    providers:
        fos_userbundle:
            id: fos_user.user_provider.username

    acl:
        connection: default

    encoders:
        FOS\UserBundle\Model\UserInterface: sha512

    firewalls:
        main:
            pattern: ^/
            form_login:
                provider: fos_userbundle
                csrf_provider: form.csrf_provider
            logout:       true
            anonymous:    true

    access_control:
        - { path: ^/login$, role: IS_AUTHENTICATED_ANONYMOUSLY }
        - { path: ^/register, role: IS_AUTHENTICATED_ANONYMOUSL
        - { path: ^/resetting, role: IS_AUTHENTICATED_ANONYMOUS
        - { path: ^/admin/, role: ROLE_ADMIN }

    role_hierarchy:
        ROLE_ADMIN:       ROLE_USER
        ROLE_SUPER_ADMIN: [ROLE_USER, ROLE_ADMIN, ROLE_ALLOWED_T
```

Add this section at the end of config.yml :

```
#config.yml
fos_user:
    db_driver: orm
    firewall_name: main
    user_class: Tech\NewsBundle\Entity\User
    profile:  # Authentication Form
        form:
            type:               fos_user_profile
            #handler:            fos_user.profile.form.handler.
            name:               fos_user_profile_form
            #validation_groups:  [Authentication] # Please note :
```

Now configure FOSUB's routing :

```
#routing.yml
fos_user_security:
    resource: "@FOSUserBundle/Resources/config/routing/security

fos_user_profile:
    resource: "@FOSUserBundle/Resources/config/routing/profile.
    prefix: /profile

fos_user_register:
    resource: "@FOSUserBundle/Resources/config/routing/registra
    prefix: /register

fos_user_resetting:
    resource: "@FOSUserBundle/Resources/config/routing/resettin
    prefix: /resetting

fos_user_change_password:
    resource: "@FOSUserBundle/Resources/config/routing/change_p
    prefix: /profile
```

Update your database schema and that's all for FOSUserBundle.

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

b. AdminBundle

→ Official documentation

composer.json :

```
#composer.json
        "sonata-project/admin-bundle": "2.2.*@dev",
        "sonata-project/block-bundle": "dev-master",
        "sonata-project/cache-bundle": "dev-master",
        "sonata-project/jquery-bundle": "1.8.*@dev",
        "sonata-project/easy-extends-bundle" : "dev-master",
        "sonata-project/intl-bundle" : "dev-master"
```

and update.
If you're getting

> sonata-project/intl-bundle dev-master requires ext-intl * -> the requested PHP extension intl is missing from your system.

You'll need to install php intl extension :
Install php-intl for your system and uncomment `extension=intl.so` in your php.ini (and restart apache).

Activate these new bundles in AppKernel :

```
//AppKernel.php
```

```
            new Sonata\BlockBundle\SonataBlockBundle(),
            new Sonata\CacheBundle\SonataCacheBundle(),
            new Sonata\jQueryBundle\SonatajQueryBundle(),
            new Sonata\EasyExtendsBundle\SonataEasyExtendsBundle(
            new Sonata\AdminBundle\SonataAdminBundle(),
            new Sonata\IntlBundle\SonataIntlBundle(),
```

config.yml : First, modify the doctrine section to have :

```
#config.yml
doctrine:
    dbal:
        #...
        types:
            json: Sonata\Doctrine\Types\JsonType
    orm:
        auto_generate_proxy_classes: %kernel.debug%
        #remove the auto_mapping: true which was right here, we
        entity_managers:
            default:
                auto_mapping: true
                mappings:
                    FOSUserBundle: ~
```

and also add these sections somewhere :

```
#config.yml
sonata_admin:
    title:      Tech News Admin Panel !
    templates:
        ## default global templates
        layout:  SonataAdminBundle::standard_layout.html.twig
        ajax:    SonataAdminBundle::ajax_layout.html.twig

        ## default actions templates, should extend a global tem
        list:    SonataAdminBundle:CRUD:list.html.twig
        show:    SonataAdminBundle:CRUD:show.html.twig
        edit:    SonataAdminBundle:CRUD:edit.html.twig
        history: SonataAdminBundle:CRUD:history.html.twig
        preview: SonataAdminBundle:CRUD:preview.html.twig
        delete:  SonataAdminBundle:CRUD:delete.html.twig

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:
```

And finally some routes to add to routing.yml :

```
#routing.yml
#SonataAdminBundle
admin:
    resource: '@SonataAdminBundle/Resources/config/routing/sona
    prefix: /admin

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

c. UserBundle

→ Official documentation

Now we will bind SonataAdminBundle to FOSUserBundle with SonataUserBundle :

composer.json :

```
#composer.json
        "sonata-project/doctrine-orm-admin-bundle": "dev-master'
        "sonata-project/user-bundle": "dev-master"
```

update your project and register these new bundles :

```
//AppKernel.php
            new Knp\Bundle\MenuBundle\KnpMenuBundle(),
            new Sonata\DoctrineORMAdminBundle\SonataDoctrineORMAc
            new Sonata\UserBundle\SonataUserBundle('FOSUserBundle
```

Add some default config : (notice that sonata_user:class:user holds our previously created User entity)

```
#config.yml
doctrine:
    orm:
        # [...]
        entity_managers:
            default:
                # [...]
                mappings:
                   # [...]
                    SonataUserBundle: ~
sonata_user:
    security_acl:     false
    class:
        user:         Tech\NewsBundle\Entity\User
    admin:                  # Admin Classes
        user:
            class:          Sonata\UserBundle\Admin\Entity\UserAc
            controller:     SonataAdminBundle:CRUD
            translation:    SonataUserBundle

        group:
            class:          Sonata\UserBundle\Admin\Entity\GroupA
            controller:     SonataAdminBundle:CRUD
            translation:    SonataUserBundle
    profile:  # Profile Form (firstname, lastname, etc ...)
        form:
            type:               sonata_user_profile
            handler:            sonata.user.profile.form.handle
            name:               sonata_user_profile_form
            validation_groups:  [Profile]
```

Some routing :

```
#routing.yml
#SonataUserBundle
sonata_user:
    resource: '@SonataUserBundle/Resources/config/routing/admin_
    prefix: /admin
```

security.yml : replace `role_hierarchy` section by :

```
#security.yml
security:
    #[...]
    role_hierarchy:
        ROLE_ADMIN:       [ROLE_USER, ROLE_SONATA_ADMIN]
        ROLE_SUPER_ADMIN: [ROLE_ADMIN, ROLE_ALLOWED_TO_SWITCH]
```

replace sections `firewalls` and `access_control` by these :

```
#security.yml
security:
    #[...]
    firewalls:
        # -> custom firewall for the admin area of the URL
        admin:
            pattern:      /admin(.*)
            form_login:
                provider:       fos_userbundle
                login_path:     /admin/login
                use_forward:    false
                check_path:     /admin/login_check
                failure_path:   null
            logout:
                path:           /admin/logout
            anonymous:    true
        # -> end custom configuration

        # defaut login area for standard users
        main:
            pattern:      .*
            form_login:
                provider:       fos_userbundle
                csrf_provider:  form.csrf_provider
                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 anc
        - { path: ^/_wdt, role: IS_AUTHENTICATED_ANONYMOUSLY }
        - { path: ^/_profiler, role: IS_AUTHENTICATED_ANONYMOUS
        - { path: ^/login$, role: IS_AUTHENTICATED_ANONYMOUSLY 

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

        - { path: ^/register, role: IS_AUTHENTICATED_ANONYMOUSL
        - { path: ^/resetting, role: IS_AUTHENTICATED_ANONYMOUS

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

Now that FOSUserBundle and SonataUserBundle are installed and configured, we have to extend SonataUserBundle with :

```
⚲ php app/console sonata:easy-extends:generate SonataUserBundle --
dest=src
```

Register this bundle in AppKernel :

```
 new Application\Sonata\UserBundle\ApplicationSonataUserBundle()
```

enable it in *config.yml* :

```
doctrine:
    orm:
        entity_managers:
            default:
                mappings:
                    ApplicationSonataUserBundle: ~
```

and modify our own User entity to herit from this by replacing in *tech/src/Tech/NewsBundle/Entity/User.php* :

```
use FOS\UserBundle\Entity\User as BaseUser;
```

by

```
use Sonata\UserBundle\Entity\BaseUser as BaseUser;
```

d. MediaBundle

→ Official documentation

I said I wanted my news authors to be able to upload pictures and files with their news, so let's install SonataMediaBundle, last step of this already very lengthy install process :

composer.json :

```
#composer.json
        "sonata-project/media-bundle" : "dev-master",
```

composer update,

```
//AppKernel.php
            new Sonata\MediaBundle\SonataMediaBundle(),
```

```
#routing.yml
#SonataMediaBundle
gallery:
    resource: '@SonataMediaBundle/Resources/config/routing/galle
    prefix: /media/gallery

```

```
media:
    resource: '@SonataMediaBundle/Resources/config/routing/medic
    prefix: /media
```

```
#config.yml
doctrine:
    orm:
        entity_managers:
            default:
                mappings:
                    SonataMediaBundle: ~

sonata_media:
    default_context: default
    db_driver: doctrine_orm # or doctrine_mongodb
    contexts:
        default:  # the default context is mandatory
            providers:
                - sonata.media.provider.dailymotion
                - sonata.media.provider.youtube
                - sonata.media.provider.image
                - sonata.media.provider.file

            formats:
                small: { width: 100 , quality: 70}
                big:   { width: 500 , quality: 70}

    cdn:
        server:
            path: /uploads/media # http://media.sonata-project.

    filesystem:
        local:
            directory:  %kernel.root_dir%/../web/uploads/media
            create:     false
```

Same as SonataUserBundle, we need to extend SonataMediaBundle as following :

```
⚘ php app/console sonata:easy-extends:generate SonataMediaBundle --
dest=src
```

Register in AppKernel :

```
//AppKernel.php
            new Application\Sonata\MediaBundle\ApplicationSonataM
```

Register in config.yml :

```
#config.yml
doctrine:
    orm:
        entity_managers:
            default:
                mappings:
                    ApplicationSonataMediaBundle: ~
```

Now we create our upload directories and chmod them correctly :

```
mkdir web/uploads && mkdir web/uploads/media && chmod -R 0777
web/uploads
```

Done !

e. Wrapping up

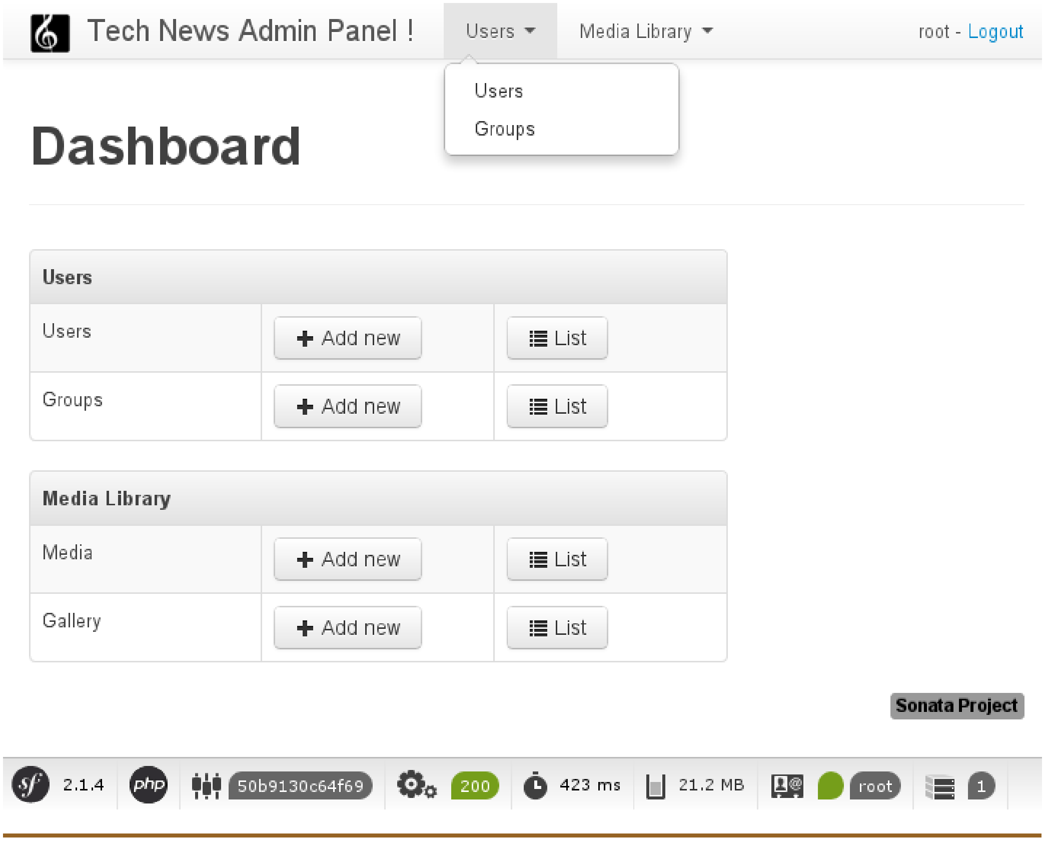
After all this, we should be all set. Well, after having updated the database, installed all assets and cleared the cache :

```
⚘ php app/console doctrine:schema:update --dump-sql
⚘ php app/console doctrine:schema:update --force
⚘ php app/console assets:install web
⚘ php app/console cache:clear
```

Create an admin user account and enjoy your brand new admin dashboard :

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

→ http://localhost/tech/web/app_dev.php/admin/dashboard



Dashboard

Users		
Users	+ Add new	List
Groups	+ Add new	List

Media Library		
Media	+ Add new	List
Gallery	+ Add new	List

1. Symfony 2.3 : ACL, SonataAdminBundle, SonataMediabundle, SonataUserBundle and FOSUserBundle
2. **Symfony 2.3, SonataAdminBundle, SonataMediabundle, SonataUserBundle and FOSUserBundle : a good project base**
3. Symfony 2.3 : Customizing SonataAdminBundle

4. Symfony 2.3, SonataAdminBundle, FOSUserBundle and ACLs

Tags :

2 comments

★ 0

Leave a message...

Best Community Share ⚙

sagitta • 21 days ago

hi, Nice tutorial and thank you for your labour.

But i just got a trouble triying to follow what was written, now it just show me the error

[Symfony\Component\Config\Exception\FileLoaderLoadException]

Cannot import resource "/var/www/L/app/config/config.yml" from "/var/www/L/app/config/config_dev.yml". (Unable to parse at line 55 (near "

entity_managers:").)

[Symfony\Component\Yaml\Exception\ParseException]

Unable to parse at line 55 (near " entity_managers:").

here is my config.yml: http://pastebin.com/uKmhPmLh

thank you

Reply Share ›

James • 25 days ago

How did you get the Tabs and the Users, Media Library?

Reply Share ›

Loosely based on a design by Wpshower