

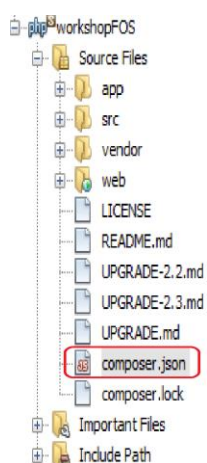
Workshop n°3 : FOS UserBundle

Objectif

Le but de ce workshop est l'installation de FOS UserBundle destiné à la gestion des utilisateurs d'une application web.

<https://github.com/FriendsOfSymfony/FOSUserBundle/blob/master/Resources/doc/index.md>

1. La première étape consiste à décommenter la variable l'extension **php_openssl.dll** dans **C:\wamp\bin\php\php5.4.16\php.ini**.
Openssl : une bibliothèque qui possède une dépendance à l'exécution, dans notre cas elle va autoriser le téléchargement de composer.
2. Créer un nouveau projet sous le répertoire : **C:\wamp\www\workshopFOS**.
3. Ajouter FOSUserBundle au fichier **composer.json**:
Composer : est une bibliothèque de gestion de dépendances pour PHP qui sert à télécharger les bundles externe



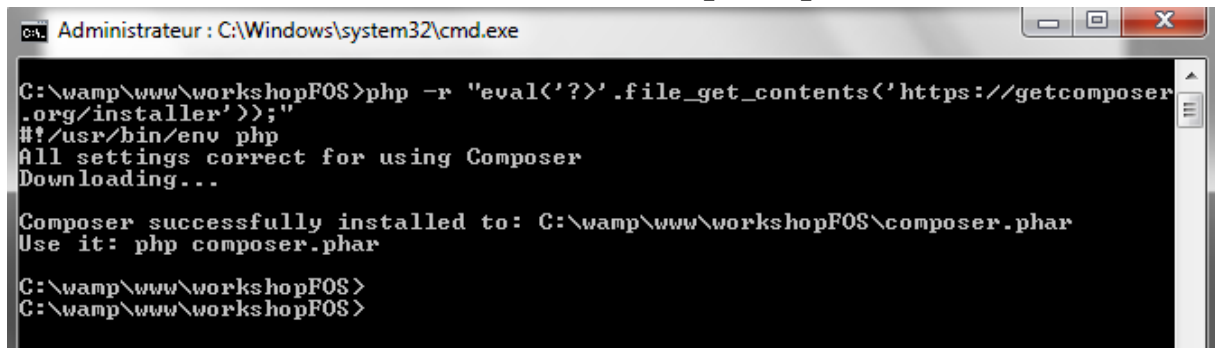
```
4  "type": "project",
5  "description": "The \"Symfony Standard Edition\" distribution",
6  "autoload": {
7      "psr-0": { "": "src/" }
8  },
9  "require": {
10     "php": ">=5.3.3",
11     "symfony/symfony": "2.3.*",
12     "doctrine/orm": ">=2.2.3,<2.4-dev",
13     "doctrine/doctrine-bundle": "1.2.*",
14     "twig/extensions": "1.0.*",
15     "symfony/assetic-bundle": "2.3.*",
16     "symfony/swiftmailer-bundle": "2.3.*",
17     "symfony/monolog-bundle": "2.3.*",
18     "sensio/distribution-bundle": "2.3.*",
19     "sensio/framework-extra-bundle": "2.3.*",
20     "sensio/generator-bundle": "2.3.*",
21     "incenteev/composer-parameter-handler": "~2.0",
22     "friendsofsymfony/user-bundle": "~2.0@dev"
23 }
```

"friendsofsymfony/user-bundle": "~2.0@dev"

Ensuite, il faut ouvrir une fenêtre de commande et se placer à la racine du projet
Et copier la commande suivante :

php -r "eval('?'>'.file_get_contents('https://getcomposer.org/installer'))';"

(vous aller avoir maintenant un fichier composer.phar sous le dossier)



```
Administrateur : C:\Windows\system32\cmd.exe

C:\wamp\www\workshopFOS>php -r "eval('?'>'.file_get_contents('https://getcomposer.org/installer'))';"
#!/usr/bin/env php
All settings correct for using Composer
Downloading...

Composer successfully installed to: C:\wamp\www\workshopFOS\composer.phar
Use it: php composer.phar

C:\wamp\www\workshopFOS>
C:\wamp\www\workshopFOS>
```

Si tout se passe bien la récupération du fichier devrait se lancer.

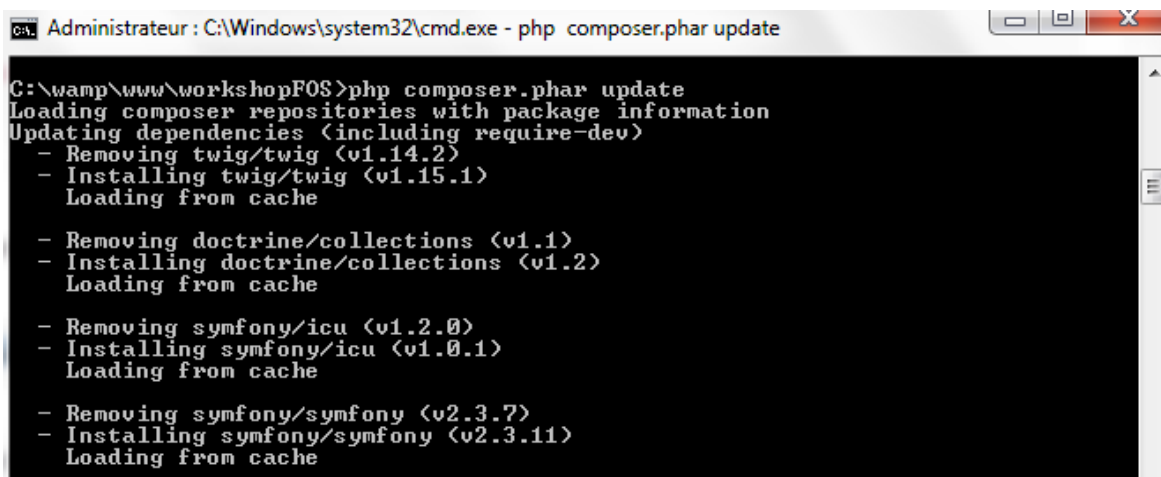
Une fois le fichier est récupéré, on lance le téléchargement des bundles depuis le terminal avec la commande suivante :

php composer.phar update

Ou bien

php -r "readfile('https://getcomposer.org/installer');" | php

→Mettre à jours la bibliothèque composer pour avoir la dernière version.



```
Administrateur : C:\Windows\system32\cmd.exe - php composer.phar update

C:\wamp\www\workshopFOS>php composer.phar update
Loading composer repositories with package information
Updating dependencies (including require-dev)
- Removing twig/twig (v1.14.2)
- Installing twig/twig (v1.15.1)
  Loading from cache

- Removing doctrine/collections (v1.1)
- Installing doctrine/collections (v1.2)
  Loading from cache

- Removing symfony/icu (v1.2.0)
- Installing symfony/icu (v1.0.1)
  Loading from cache

- Removing symfony/symfony (v2.3.7)
- Installing symfony/symfony (v2.3.11)
  Loading from cache
```

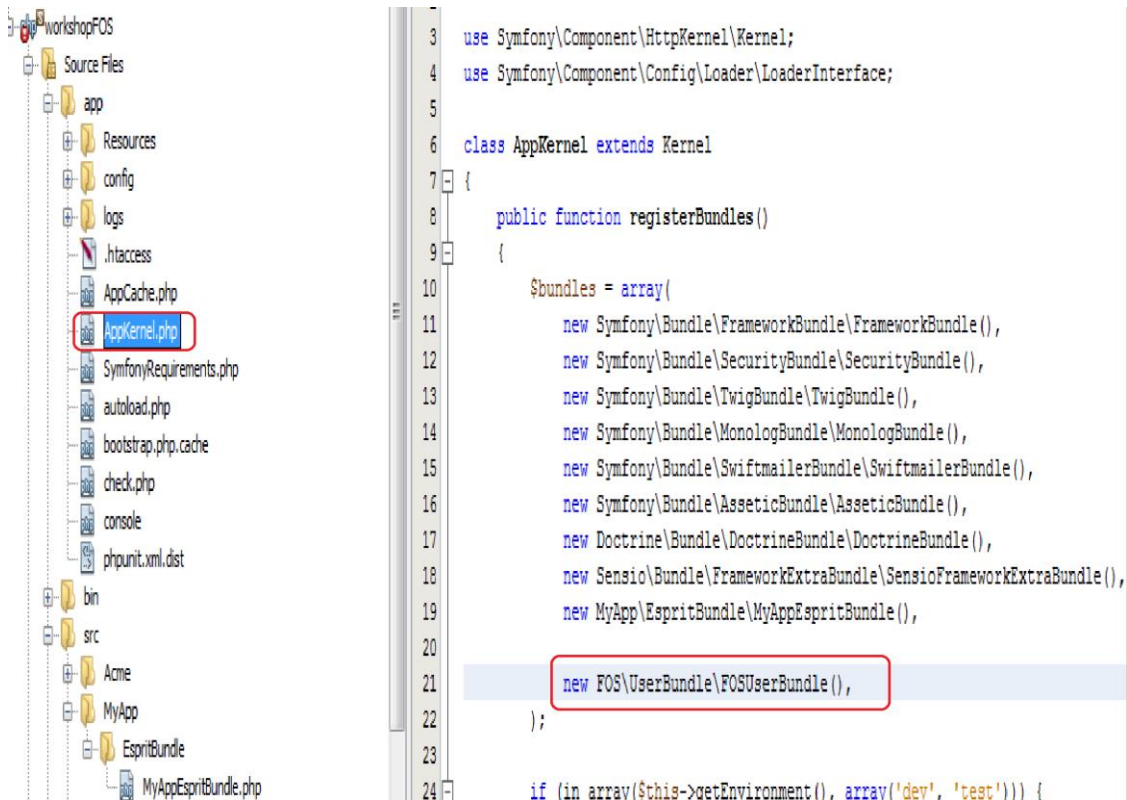
php composer.phar update friendsofsymfony/user-bundle

→ Télécharger le bundle fosUserBundle

Une fois que le téléchargement est effectué avec succès, nous allons trouver dans le dossier vendor : notre bundle friendsofsymfony.

4. L'étape suivante consiste à créer un bundle : **MyApp\UserBundle**.

5. Activer le bundle dans le kernel : **new FOS\UserBundle\FOSUserBundle()**,



6. Une fois que notre bundle est prêt à être utiliser, nous créons un dossier **Entity** sous **UserBundle** puis **User.php** :

```
<?php
namespace MyApp\UserBundle\Entity;
use FOS\UserBundle\Model\User as BaseUser;
use Doctrine\ORM\Mapping as ORM;

/**
 * @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
    }
}
?>

```

7. Afin d'assurer la sécurité de l'application, nous devons alors configurer notre **App\Config\security.yml**

```

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

    role_hierarchy:
        ROLE_ADMIN:       ROLE_USER
        ROLE_SUPER_ADMIN: ROLE_ADMIN

    providers:
        fos_userbundle:
            id: fos_user.user_provider.username

    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_ANONYMOUSLY }
        - { path: ^/resetting, role: IS_AUTHENTICATED_ANONYMOUSLY }
        - { path: ^/admin/, role: ROLE_ADMIN }

```

Firewalls: système de sécurité du projet, détermine si un utilisateur doit ou ne doit pas être authentifié

Pour plus de détails voir ici :

<http://symfony.com/fr/doc/current/book/security.html>

8. La configuration de notre bundle se fait dans le fichier **App\Config.yml**.

```
#UserBundle:
fos_user:
  db_driver: orm # other valid values are 'mongodb', 'couchdb' and 'propel'
  firewall_name: main
  user_class: MyApp\UserBundle\Entity\User

fos_user:
  db_driver: orm # other valid values are 'mongodb', 'couchdb' and 'propel'
  firewall_name: main
  user_class: MyApp\UserBundle\Entity\User
```

9. La configuration des routing se fait dans : **App\routing.yml**

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

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

fos_user_register:
  resource: "@FOSUserBundle/Resources/config/routing/registration.xml"
  prefix: /register

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

fos_user_change_password:
  resource: "@FOSUserBundle/Resources/config/routing/change_password.xml"
  prefix: /profile
```

Ou bien :

Importer les fichiers de routing de FosUserBundle.

```
fos_user:
    resource: "@FOSUserBundle/Resources/config/routing/all.xml"
```

```
fos_user:
    resource: "@FOSUserBundle/Resources/config/routing/all.xml"
```

10. Configurer notre base de données :

Donner le nom de la base de données dans le fichier parameters.yml

```
Php app/console doctrine:database:create
php app/console doctrine:schema:create
```

11. Personnaliser les formulaires d'inscription et d'authentification en français.

Pour changer et personnaliser l'interface d'authentification ainsi que l'interface d'inscription.

a- Dans le fichier **parameters.yml** changer la langue en anglais :

```
parameters:
    database_driver: pdo_mysql
    database_host: 127.0.0.1
    database_port: null
    database_name: TestFosUser
    database_user: root
    database_password: null
    mailer_transport: smtp
    mailer_host: 127.0.0.1
    mailer_user: null
    mailer_password: null
    locale: fr
    secret: ThisTokenIsNotSoSecretChangeIt
```

b- Dans le fichier **config.yml**, décommenter la ligne de traduction.

framework:

#esi:

translator: { **fallback:** "%locale%" }

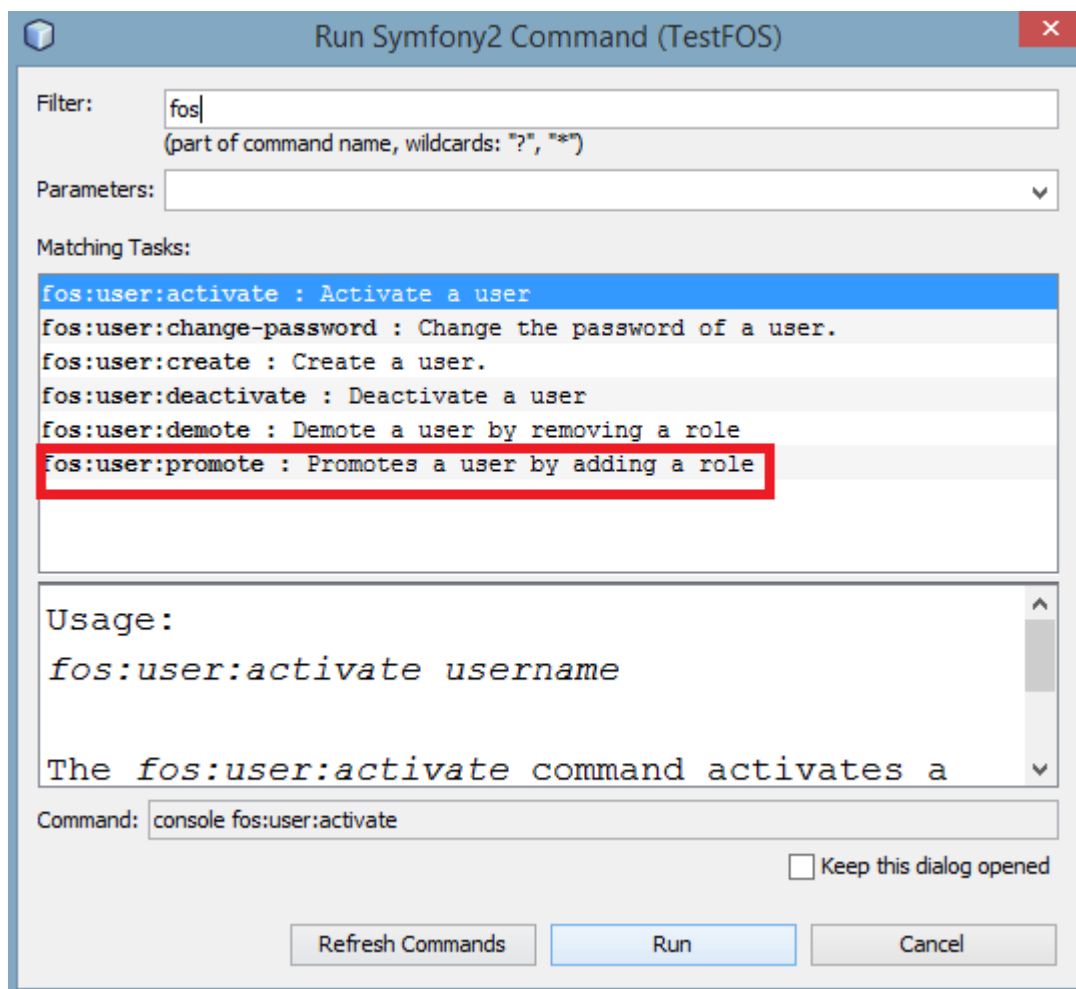
On va tester si notre bundle a été installé avec succès en tapant

l'adresse :

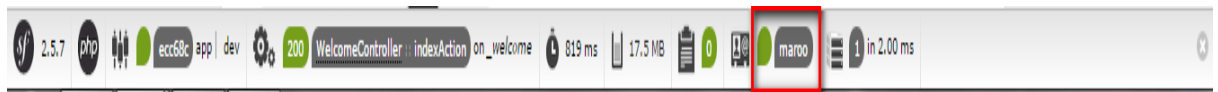
localhost/workshopFOS/web/app_dev.php/register : pour l'inscription

localhost/workshopFOS/web/app_dev.php/login :pour l'authentification

12.Création des nouveaux rôles et affectation des rôles pour les utilisateurs.



Supposons qu'on a ajouté le rôle admin pour l'utilisateur inscrit, ci-dessous une image pour voir si le rôle a été bien attribué.



En cliquant sur le nom d'utilisateur on pourra voir le changement :

Security	
Username	maroo
Authenticated?	yes
Roles	[ROLE_ADMIN, ROLE_USER]
Token class	Symfony\Component\Security\Core\Authentication\Token\UsernamePasswordToken

13. Les contrôles d'accès

Vous pouvez créer vos pages et les restreindre en modifiant le fichier **app/config/security.yml** en agissant sur la partie **access_control** (en supposant bien sûr que vous avez déjà créé une nouvelle route nommée ListProduit)

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

14. Changer la classe de login

- créer une entité (par exemple étudiant) sous `UserBundle\Entity\` qui hérite de la classe `BaseUser`, étudiant contiendra les champs: id, nom, et prenom.
- supprimer l'entité `User` sous `UserBundle\Entity`
- Mettre à jour de la table

d- Faire le changement au niveau du fichier config.yml comme suite :

```
fos_user:
  db_driver: orm # other valid values are 'mongodb', 'couchdb' and 'propel'
  firewall_name: main
  user_class: MyApp\UserBundle\Entity\Etudiant
```

e- Pour Ajouter les champs, nom et prenom dans le formulaire d'inscription, aller à :

vendor\friendsofsymfony\UserBundle\Form\Type\RegistrationFormType.php

15. Intégration d'un Template a FosUserBundle

a- Surcharger fosuserBundle

Ajouter la fonction getParent à la classe : EspritUserBundle

```
public function getParent()
{
    return 'FOSUserBundle';
}
```

a- Copiez vos dossiers **CSS, images** et **js** sous le répertoire **Web/**

b- Créez une nouvelle vue **layout.html.twig** sous le répertoire

Resources/views/ de votre Bundle

c- Copier le contenu de la page index de votre template dans le fichier
layout.html.twig

d- Utiliser **asset** pour faire appel à nos images et au fichier css.

e- Réserver un block dans la page layout.html.twig pour les formulaires
d'inscription et d'authentification en utilisant : **{% block formulaire
%}**

f- Copier tous les dossiers qui se trouvent sous : vendor\
friendsofsymfony\user-bundle\ Resources\views, dans votre nouveau
Bundle (**UserBundle**) sous Resources\views.

- g- On va tester l'intégration de template sur la page de login :
Dans le fichier login.html.twig, ajouter le nécessaire pour pouvoir hériter de la page layout.html.twig

```
{% extends "MyAppUserBundle::layout.html.twig" %}
```

```
{% trans_default_domain 'FOSUserBundle' %}
```

```
{% block formulaire %}
```

```
{% if error %}
```

```
    <div>{{ error|trans }}</div>
```

```
{% endif %}
```

```
<form action="{{ path('fos user security check') }}"
```

Donc vous allez avoir comme résultat en tapant l'adresse :
localhost/workshopFOS/web/app_dev.php/login

Nom d'utilisateur : Mot de passe : ☐ Se souvenir de moi

Home Link two Link three Link four Link five

Sub Menu

- Home
- Products
- What we do
- Contact us
- Privacy Policy
- Links