DOCUMENTATION TECHNIQUE POUR DEVELOPPEUR JUNIOR SYMFONY

Étape 1 :

Commencez par créer un dossier puis à y cloner le repository de l’application en tapant dans le CLI de votre IDE : git clone https://github.com/saro0h/projet8-TodoList.git

Étape 2 :

Puis, placez vous à la racine du projet.

Tapez : composer install

Tous les composants nécessaires au fonctionnement de l’application web vont s’installer.

Étape 3 :

Ensuite tapez : symfony serve

Vous devez voir ce message apparaitre :

[OK] Web server listening

The Web server is using PHP FPM 7.3.19

http://127.0.0.1:8000

Étape 4 :

Rendez vous sur cette adresse http://127.0.0.1:8000

On se retrouve sur la page suivante : <http://localhost:8000/login>

Dans le profiler, on peut voir la version de Symfony utilisé par l’application web.

Il s’agit de la version 3.1.6.

Étape 5 :

Commençons par upgrade cette version vers la version 3.4

A la racine du projet se trouve Composer.Json . Ouvrez le et dans la section require : Changez la version de Symfony/symfony par : 3.4.\*

Puis dans le CLI, tapez : composer update

Étape 6 :

Redemarrez le server et rendez vous sur <http://localhost:8000/login>

Dans le profiler, on peut voir que la version de symfony utilisé est désormais la 3.4.42

Notre but est d’arriver à la version 4 de Symfony.

Mais pour cela, nous devons corrigez les dépreciations.

Étape 7 :

Dans la barre du profiler de Symfony, cliquez sur le bouton pour voir la liste des dépréciations. Il y en a 11.

Étape 8:

Dépréciation  : User Deprecated: Symfony\Component\HttpKernel\Kernel::loadClassCache() is deprecated since Symfony 3.3, to be removed in 4.0.

On peut voir en cliquant sur Trace que la fonction qui est dépréciée se trouve dans web/app\_dev.php .

Ouvrez le fichier et supprimer la ligne 26 :

$kernel->loadClassCache();

Ouvrez aussi web/app.php et supprimez la ligne 10 :

$kernel->loadClassCache();

Si l’on refresh la page, on peut voir qu’il n’y a plus que 9 dépréciations.

Étape 9 :

Dépréciation : Using the unquoted scalar value "**!event**" is deprecated since Symfony 3.3 and will be considered as a tagged value in 4.0. You must quote it in "**/Users/jvjlondon/Desktop/SoutenanceP8/projet8-TodoList/app/config/config\_dev.yml**" on line 20.

Pour corriger celle ci, rendez vous dans app/config/config\_dev.yml et ajoutez des « » autour des valeurs des lignes 20 et 23 comme ceci :

channels: ["!event"]  
console:  
 type: console  
 channels: ["!event", "!doctrine"]

Étape 10 :

Dépréciations : The "**framework.trusted\_proxies**" configuration key has been deprecated in Symfony 3.3. Use the Request::setTrustedProxies() method in your front controller instead.

Pour corriger, celle ci rendez vous dans config/config.yml et supprimez la ligne 26 :

trusted\_proxies: ~

Étape 11 :

Dépréciations :

|  |  |
| --- | --- |
|  | Not setting "**logout\_on\_user\_change**" to true on firewall "**main**" is deprecated as of 3.4, it will always be true in 4.0.  Pour corriger celle ci, allez dans app/config/security.yml et ajoutez cette ligne la dans la section main :  logout\_on\_user\_change: true  Cela devrait être la ligne 17. |

Étape 12 :

Il reste 5 dépréciations

Pour corriger une partie d’entre elles, nous allons nous rendre sur cette adresse et comparer notre composer.json à celui de la version standard de Symfony 3.4

<https://github.com/symfony/symfony-standard/blob/3.4/composer.json>

Donc, nous allons changer ces bundles suivants :

"symfony/swiftmailer-bundle": "^2.6.4"

"symfony/monolog-bundle": "^3.1.0",

"sensio/framework-extra-bundle": "^5.0.0",

"sensio/distribution-bundle": "^5.0.19",

Ensuite à la racine du projet, dans le CLI, tapez : composer update

Étape 13 :

Plusieurs nouvelles dépréciations sont apparues, mais en contrepartie nous sommes à jour dans la version de nos bundles.

Voyons une autre dépréciations :

User Deprecated: The "**Sensio\Bundle\FrameworkExtraBundle\Configuration\Route**" annotation is deprecated since version 5.2. Use "**Symfony\Component\Routing\Annotation\Route**" instead.

Pour corriger celle-ci, rendez vous dans src/AppBundle/Controller et remplacez dans tous les controller :

use Sensio\Bundle\FrameworkExtraBundle\Configuration\Route;

par :

use Symfony\Component\Routing\Annotation\Route;

Étape 14 :

Voyons une autre dépréciations :

The "**framework.trusted\_proxies**" configuration key has been deprecated in Symfony 3.3. Use the Request::setTrustedProxies() method in your front controller instead.

Pour la corriger, rendez vous dans un premier temps dans app/config/config.yml et supprimez la ligne 26 :

trusted\_proxies: ~

Puis allez dans web/app.php et ajoutez la ligne suivante :

Request::setTrustedProxies()

en dessous de la ligne 9 :

$kernel = new AppKernel('prod', false);  
Request::*setTrustedProxies*()

Étape 15 :

Voyons une autre dépréciations :

User Deprecated: Creating Doctrine\ORM\Mapping\UnderscoreNamingStrategy without making it number aware is deprecated and will be removed in Doctrine ORM 3.0.

Pour corriger celle-ci, rendez vous dans app/config/config.yml et changez la ligne 60 par :

naming\_strategy: doctrine.orm.naming\_strategy.underscore\_number\_aware

Étape 16 :

Voyons la dernière dépréciation :

|  |  |
| --- | --- |
|  | Enabling the "**sensio\_framework\_extra.router.annotations**" configuration is deprecated since version 5.2. Set it to false and use the "**Symfony\Component\Routing\Annotation\Route**" annotation from Symfony itself. |

Pour la corriger rendez vous dans app/config/config.yml et ajoutez ces lignes de codes à la fin :

sensio\_framework\_extra:  
 router:  
 annotations: false

Voila il n’y a plus aucune dépréciations.

Nous allons pouvoir upgrade la version de Symfony à la version 4.

Étape 17 :

Rendez vous dans composer.json et changez dans require :

"symfony/symfony": "^4.0",

Et aussi dans require-dev :

"symfony/phpunit-bridge": "^4.0"

Puis supprimer dans require :

"incenteev/composer-parameter-handler": "^2.0",

"symfony/swiftmailer-bundle": "",

Et supprimer aussi :

"sensio/distribution-bundle": "^5.0.19",

Et dans require-dev :

Supprimer :

"sensio/generator-bundle": "^3.0",

Ensuite dans le CLI, depuis la racine du projet tapez : composer update

Étape 18 :

Nous sommes maintenant sur Symfony 4 ! Super !

Nous allons pouvoir installer la structure Flex :

Son but ? **Automatiser l’installation et la suppression de vos dépendances** en fournissant une **configuration par défaut** sans avoir à aller lire la doc pour trouver quelle configuration écrire, quelles routes charger ou autre tâche rébarbative à effectuer. Et dès Symfony 4.0, Flex sera le moyen par défaut pour développer une application Symfony.

Commencez par vérifier que votre composer est à la derniere version possible.

Ensuite installez flex : composer require symfony/flex

Vidons désormais notre dossier vendor pour le reremplir avec et configurer sa librairie avec flex.

Dans le CLI, tapez : rm -rf vendor

Puis : composer install

Étape 19 :

Rendez vous dans composer.json et dans require supprimez :

Symfony/symfony

Et ajoutez :

"php": "^7.2",  
"symfony/console": "^4.0",  
"symfony/framework-bundle": "^4.0",  
"symfony/lts": "^4@dev",  
"symfony/yaml": "^4.0"

Puis dans require-dev, ajoutez :

"symfony/dotenv": "^4.0",

Puis tout en haut de composer.json, ajoutez les lignes suivantes :

"config": {  
 "preferred-install": {  
 "\*": "dist"  
 },  
 "sort-packages": true  
},

Puis remplacez la section scripts et extra par celles-ci :

"scripts": {  
 "auto-scripts": {  
 "cache:clear": "symfony-cmd",  
 "assets:install --symlink --relative %PUBLIC\_DIR%": "symfony-cmd"  
 },  
 "post-install-cmd": [  
 "@auto-scripts"  
 ],  
 "post-update-cmd": [  
 "@auto-scripts"  
 ]  
},  
"conflict": {  
 "symfony/symfony": "\*"  
},  
"extra": {  
 "symfony": {  
 "allow-contrib": true  
 }  
}

Remplacez les lignes suivantes aussi , pour autoload et autoload-dev

|  |
| --- |
|  |

"autoload": {  
 "psr-4": {  
 "App\\": "src/"  
 }  
},  
"autoload-dev": {  
 "psr-4": {  
 "App\\Tests\\": "tests/"  
 }  
},

Puis tapez dans le CLI, composer update .

Étape 20 :

Allez à l’adresse suivante :

<https://github.com/symfony/recipes/blob/master/symfony/console/4.4/bin/console>

Puis copiez le contenu et coller le dans notre projet dans bin/console .

Étape 21 :

Nous allons migrer tous le contenu de app/ vers la nouvelle strucuture symfony4 petit à petit.

On peut s’occuper de nouveaux composants pour commencer avec les formulaires et validator

Tapez dans le CLI :

composer require form

et

composer require validator

Grâce à flex tout est déjà configurer pour nous.

Passons à la suite.

Nous allons installer le composant de sécurité pour crsf\_protection

Tapez dans le CLI :

composer require security\_csrf

Étape 22 :

Ensuite occupons nous de twig :

Composer require twig

Un nouveau dossier template a été créer.

Migrons toutes nos anciennes vues dans ce dossier.

Pour cela, allez dans app/Resources/view, copiez tous les dossiers et collez les dans template et overwrite le base.html.twig .

Vous pouvez ensuite supprimer le dossier Resources.

Étape 23 :

Dans config/framework.yaml

Changez la ligne 14 par celle-ci :

fragments: ~

Puis voici le tour des asset, tapez dans le CLI :

Composer require asset

Étape 24 :

Allez dans src/AppBundle et copiez le contenu des dossiers Controller et Entity pour le copier dans src/Controller et src/Entity.

Bougez le dossier Form aussi dans src/

Puis supprimez src/AppBundle.

Étape 25 :

Changez tous les namespaces et les use contenant AppBundle par App dans les controllers et les entités ainsi que dans le dossier Form

Puis dans les controllers changer la classe extends par AbstractController ainsi que le use par

use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;

et supprimer celui qui n’est plus utilisé.

Étape 26 :

Migrons ensuite la sécurité en installant le bundle suivant :

Dans le CLI, tapez : composer require security

Copiez le contenu de app/config/security.yml et collez le dans config/security.yaml

En changeant par la suite la ligne 8 :

class: App:User