

Mission réalisée E4 : Montée de version d'un projet Symfony et gestion des erreurs.

<div>Compétences mises en œuvre</div> <div>Réalisations professionnelles (intitulé et liste des documents et productions associés)</div>	Période (sous la forme du JJ/MM/AA au JJ/MM/AA)	Gérer le patrimoine informatique	Répondre aux incidents et aux demandes d'assistance et d'évolution	Développer la présence en ligne de l'organisation	Travailler en mode projet	Mettre à disposition des utilisateurs un service informatique	Organiser son développement professionnel
		<ul style="list-style-type: none">• Recenser et identifier les ressources numériques• Exploiter des référentiels, normes et standards adoptés par le prestataire informatique• Mettre en place et vérifier les niveaux d'habilitation associés à un service<ul style="list-style-type: none">• Vérifier les conditions de la continuité d'un service informatique• Gérer des sauvegardes• Vérifier le respect des règles d'utilisation des ressources numériques	<ul style="list-style-type: none">• Collecter, suivre et orienter des demandes• Traiter des demandes concernant les services réseau et système, applicatifs• Traiter des demandes concernant les applications	<ul style="list-style-type: none">• Participer à la valorisation de l'image de l'organisation sur les médias numériques en tenant compte du cadre juridique et des enjeux économiques• Référencer les services en ligne de l'organisation et mesurer leur visibilité.• Participer à l'évolution d'un site Web exploitant les données de l'organisation.	<ul style="list-style-type: none">• Analyser les objectifs et les modalités d'organisation d'un projet• Planifier les activités• Évaluer les indicateurs de suivi d'un projet et analyser les écarts	<ul style="list-style-type: none">• Réaliser les tests d'intégration et d'acceptation d'un service• Déployer un service• Accompagner les utilisateurs dans la mise en place d'un service	<ul style="list-style-type: none">• Mettre en place son environnement d'apprentissage personnel• Mettre en œuvre des outils et stratégies de veille informationnelle• Gérer son identité professionnelle• Développer son projet professionnel
Réalizations en milieu professionnel en cours de seconde année							
Montée de version d'un projet symfony	09/01/2022 au 19/01/2022		X		X		X

Sommaire :

- **Quelle est ma mission ?**
- **Définition :**
 - **Qu'est-ce que Symfony ?**
- **Réalisation de la mission.**
- **Bilan**

Lors de ce second stage, l'une des missions que l'on m'a demandé de réaliser est :

- L'adaptation d'un projet symfony en version 3.4 à la version 5.4.
- La gestion des erreurs.

Sur cette documentation, je vais généraliser comment j'ai répondu à cette mission.

Tout d'abord qu'est-ce que symfony ?



Symfony est un framework PHP puissant et robuste, celui-ci offre des fonctionnalités permettant de faciliter de d'accélérer le développement.

N'ayant jamais utilisé de framework PHP, je me suis dans un premier formé à utiliser Symfony à l'aide de la documentation officielle :

<https://symfony.com/doc/current/index.html>

Après avoir compris les bases de l'utilisation de symfony, j'ai procédé aux installations complémentaires et nécessaires sur un environnement linux.

Réalisation de la mission :

J'ai installé composer à l'aide des commandes :

```
php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
php -r "if (hash_file('sha384', 'composer-setup.php') ===
'906a84df04cea2aa72f40b5f787e49f22d4c2f19492ac310e8cba5b96ac8b64115ac402c8c
d292b8a03482574915d1a8') { echo 'Installer verified'; } else { echo
'Installer corrupt'; unlink('composer-setup.php'); } echo PHP_EOL;"
```

```
php composer-setup.php
php -r "unlink('composer-setup.php');"
```

Puis j'ai installé Symfony CLI :

```
echo 'deb [trusted=yes] https://repo.symfony.com/apt/ /' | sudo tee
/etc/apt/sources.list.d/symfony-cli.list
sudo apt update
sudo apt install symfony-cli
```

Et finalement j'ai installé PostgreSQL qui est un système de gestion de base de données :

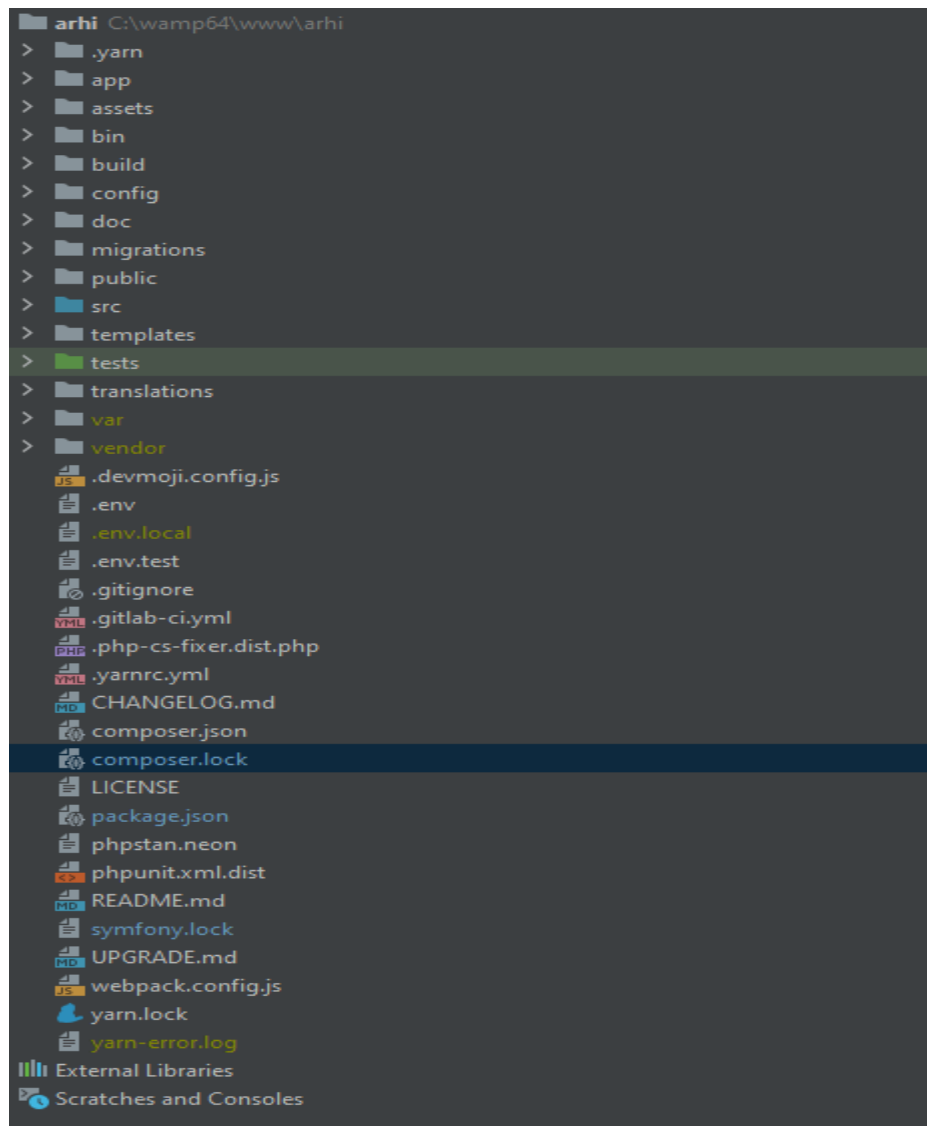
```
sudo apt install postgresql postgresql-contrib postgresql-client
```

Et puis j'ai récupéré le projet de l'entreprise sur la forge de l'intranet, ce projet est une application de gestion des ressources humaines. Afin de récupérer le projet j'ai fait un :

git-clone 'du lien https de la forge'.

```
Cloning into 'arhi'...
remote: Enumerating objects: 13813, done.
remote: Counting objects: 100% (166/166), done.
remote: Compressing objects: 100% (106/106), done.
Receiving objects: 17% (2374/13813), 2.89 MiB | 55.00 KiB/s
```

Après avoir fait l'installation j'ai un projet correspondant à cela :



J'ai par la suite fait la commande `composer update` afin de mettre les dépendances à jours.

Ensuite des problèmes de package ne supportant pas PHP 8 pouvant surgir, en fonction des messages d'erreurs indiqués je changeais ou je supprimais la version du package dans le `composer.json`.

Je regardais sur le site <https://packagist.org/> les packages à mettre à jours.

Par exemple :

doctrine/orm

⬇ composer require doctrine/orm

Object-Relational-Mapper for PHP

v2.5.0

2015-04-02 20:40 UTC

requires <ul style="list-style-type: none">• php: >=5.4• ext-pdo: *• doctrine/cache: ~1.4• doctrine/collections: ~1.2• doctrine/common: >=2.5-dev,<2.6-dev• doctrine/dbal: >=2.5-dev,<2.6-dev• doctrine/instantiator: ~1.0.1• symfony/console: ~2.5	requires (dev) <ul style="list-style-type: none">• phpunit/phpunit: ~4.0• satooshi/php-coveralls: dev-master• symfony/yaml: ~2.1	suggests <ul style="list-style-type: none">• symfony/yaml: If you want to use YAML Metadata Mapping Driver
provides <p>None</p>	conflicts <p>None</p>	replaces <p>None</p>

Nous avons doctrine/orm version 2.5 qui ne supporte que les versions php >=5.4.

Puis :

requires	requires (dev)	suggests
<ul style="list-style-type: none">• php: ^7.1 ^8.0• doctrine/annotations: ^1• doctrine/cache: ^1.11 ^2.0• doctrine/dbal: ^2.13.1 ^3.3.2• doctrine/persistence: ^2.2• doctrine/sql-formatter: ^1.0.1• symfony/cache: ^4.3.3 ^5.0 ^6.0• symfony/config: ^4.4.3 ^5.0 ^6.0• symfony/console: ^3.4.30 ^4.3.3 ^5.0 ^6.0• symfony/dependency-injection: ^4.4.18 ^5.0 ^6.0• symfony/deprecation-contracts: ^2.1 ^3• symfony/doctrine-bridge: ^4.4.22 ^5.2.7 ^6.0• symfony/framework-bundle: ^3.4.30 ^4.3.3 ^5.0 ^6.0	<ul style="list-style-type: none">• doctrine/coding-standard: ^9.0• doctrine/orm: ^2.10 ^3.0• friendsofphp/proxy-manager-lts: ^1.0• phpunit/phpunit: ^7.5 ^8.0 ^9.3 ^10.0• psalm/plugin-phpunit: ^0.16.1• psalm/plugin-symfony: ^3• psr/log: ^1.1.4 ^2.0 ^3.0• symfony/phpunit-bridge: ^5.2 ^6.0• symfony/property-info: ^4.3.3 ^5.0 ^6.0• symfony/proxy-manager-bridge: ^3.4 ^4.3.3 ^5.0 ^6.0• symfony/security-bundle: ^4.4 ^5.0 ^6.0• symfony/twig-bridge: ^3.4.30 ^4.3.3 ^5.0 ^6.0	<ul style="list-style-type: none">• ext-pdo: *• doctrine/orm: The Doctrine ORM integration is optional in the bundle.• symfony/web-profiler-bundle: To enable the data collector.

Ici nous avons la version 2.6.2 de doctrine/orm celle-ci supporte des versions PHP allant de ^7.1 aux versions PHP ^8.0.

Donc afin de réaliser la modification dans le fichier composer.json :

```
"require": {
    "php": ">=8.0",
    "a2lix/translation-form-bundle": "^3.0", 3.2.1
    "addwiki/mediawiki-api-base": "^2.8", 2.8.0
    "composer/package-versions-deprecated": "^1.8", 1.11.99.5
    "doctrine/annotations": "^1.0", 1.13.2
    "doctrine/doctrine-bundle": "^2.2", 2.6.2
    "doctrine/doctrine-migrations-bundle": "^3.0", 3.2.2
    "doctrine/orm": "~2.6", 2.12.0
    "friendsofphp/php-cs-fixer": "^3.2", v3.8.0
    "guzzlehttp/guzzle": "~6.0", 6.5.5
    "h4cc/wkhtmltopdf-amd64": "0.12.4", 0.12.4
    "incenteev/composer-parameter-handler": "^2.0", v2.1.4
    "knplabs/doctrine-behaviors": "^2.2", 2.2.0
    "liip/imaginary-bundle": "~2", 2.7.6
    "m4tthumphrey/php-gitlab-api": "^9.6", 9.19.2
    "ob/highcharts-bundle": "^1.7", 1.7
    "oomphinc/composer-installers-extender": "^2.0", 2.0.1
    "php-http/guzzle6-adapter": "^1.1", v1.1.1
    "php-translation/symfony-bundle": "^0.12.3", 0.12.6
    "rmccue/requests": ">=1.0", 2.0.2
    "sensio/framework-extra-bundle": "^6.1", v6.2.6
    "symfony/asset": "^5.4", v5.4.7
```

Je modifiais le numéro de version, et puis je mettais à jours.

Certaines fois, il fallait supprimer le composer.lock afin que les modifications se fassent.

Après avoir mis à jour les dépendances, j'ai lancé le serveur symfony à l'aide de la commande :

```
C:\wamp64\www\arhi> symfony server:start -d
```

```
[OK] Web server listening
The Web server is using PHP CGI 7.4.9
http://127.0.0.1:8000
```

Je suis ensuite allé sur le projet, auquel j'ai corrigé les erreurs que l'on m'indiquait à l'aide des indications d'erreurs de symfony.

500 → @homepag... 4867 ms 40.0 MiB 1 1572 in 1582.19 ... elliot 5 ms 4 in 9.76 ms 0 Server 5.4.7

Voici comment sont indiqués les erreurs :

```
Twig\Error
RuntimeError

in C:\wamp64\www\arhi\templates\admin\stats.html.twig (line 129)

124.             Graphiques
125.         </button>
126.     </h2>
127.     <div id="stat-chart" class="accordion-collapse collapse" aria-labelledby="stat-chart">
128.         <div class="accordion-body p-0">
129.             {{ render(controller('App\Controller\Admin\ChartsController::allCharts')) }}
130.         </div>
131.     </div>
132. </div>
133.
134. <div class="accordion-item">
```

Bilan :

Globalement cette mission m'a été bénéfique car cela m'a permis d'en apprendre sur symfony et son utilisation. D'être plus organisé afin de gérer des erreurs et une montée de version. De mieux travailler en mode projet, et de réaliser des tests unitaires.