

## Groupe 3 - Synthèse

Hugo Ligneres - Samantha Ortega - Hamza Takarrouht - Hugo Rytlewski

UE L313 - Semaine 4

---



# Table des matières

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Organisation générale</b>	<b>3</b>
2.1	Répartition du travail . . . . .	3
2.2	Méthode d'organisation . . . . .	4
<b>3</b>	<b>Fonctionnalités et défis techniques</b>	<b>4</b>
3.1	Installation de Symfony . . . . .	4
3.2	Dépôt GitHub du projet . . . . .	5
3.3	Création du CRUD . . . . .	6
3.4	Problèmes migrations . . . . .	6
3.5	Ajout de Bootstrap 5.3 au projet . . . . .	7
3.5.1	Utilisation de Bootstrap . . . . .	8
<b>4</b>	<b>Conclusion</b>	<b>9</b>

# 1 Introduction

Pour cette deuxième et dernière semaine des projets de groupe, nous devons réaliser un clone de Watson, cette fois-ci avec le framework Symfony et les différents outils associés. Le but était de mettre en place un système similaire à celui de Watson, c'est-à-dire un CRUD permettant de gérer le stockage et l'affichage des liens sur le site.

Nous allons d'abord voir comment nous nous sommes organisés pour ce travail de groupe. Puis, dans une deuxième partie, nous reviendrons sur ce que nous avons produit.

## 2 Organisation générale

### 2.1 Répartition du travail

Dès le début de la semaine, nous avons défini une répartition du travail pour être le plus efficace possible. Chaque membre avait des responsabilités différentes :

Nom	Mission
Hugo L	Installation du projet Gestion dépôt GitHub Rédaction du rapport de synthèse Rendu Twig/CSS
Samantha	Formulaire de l'ajout de liens Rendu Twig/CSS
Hamza	Pagination Amélioration de la suppression
Hugo R	Mise en place du CRUD

Répartition des rôles au sein du groupe

## 2.2 Méthode d'organisation

Pour le dépôt GitHub, nous avons utilisé la fonctionnalité des branches pour faciliter le travail en asynchrone. En début de semaine 3 de l'UE, nous avons décidé tous ensemble que ces branches devaient respecter la nomenclature suivante : "nom\_fonctionnalité". Par exemple, "hugo\_crud" ou "samantha\_formulaire\_ajout".

Malgré tout, certaines des branches du dépôt ne suivent pas cette nomenclature.

Pour la communication interne, nous avons décidé d'utiliser l'outil Teams. Nous y avons intégré un cahier de bord, pour que chaque membre puisse y noter tous les changements apportés au projet.

## 3 Fonctionnalités et défis techniques

### 3.1 Installation de Symfony

L'installation de Symfony a été réalisée en utilisant la version 6.4 [5] du framework, avec PHP 8.2+, et en utilisant le CLI Symfony. Nous avons choisi cette option plutôt que Docker pour deux raisons : l'habitude de travailler avec le CLI pour certains des membres, et les fonctionnalités qui viennent avec le CLI [4]. Le système de base de données utilisé est SQLite, dont la configuration est trouvable dans le fichier `.env.local`.

Avant de procéder à l'installation du projet, nous avons dû nous assurer que nos environnements de travail étaient prêts à utiliser Symfony et ses divers composants. Pour cela, l'astuce était de se référer au livre officiel [3] de Symfony. En effet, dans l'introduction de celui-ci, il est fourni une commande permettant de vérifier rapidement si tous les éléments qui seront utilisés durant la lecture du livre sont installés ou non. Bien sûr, pour notre projet, nous n'avions pas besoin de tous ces éléments, mais cela nous permettait de vérifier en quelques secondes si l'essentiel était installé.

```
Welcome to fish, the friendly interactive shell
Type help for instructions on how to use fish
administrateur@pc-hugo -> symfony book:check-requirements
[OK] Git installed
[OK] PHP installed version 8.3.14 (/usr/bin/php8.3)
[OK] PHP extension "json" installed - required
[OK] PHP extension "ctype" installed - required
[KO] PHP extension "pdo_pgsql" not found, please install it - required
[OK] PHP extension "zip" installed - optional - needed only for chapter 17 (Panther)
[OK] PHP extension "amqp" not found, optional - needed only for chapter 32
[OK] PHP extension "iconv" installed - required
[OK] PHP extension "session" installed - required
[OK] PHP extension "gd" installed - optional - needed only for chapter 23 (Imagine)
[OK] PHP extension "redis" not found, optional - needed only for chapter 31
[OK] PHP extension "tokenizer" installed - required
[OK] PHP extension "xml" installed - required
[KO] PHP extension "intl" not found, please install it - required
[OK] PHP extension "xsl" installed - required
[OK] PHP extension "curl" installed - optional - needed only for chapter 17 (Panther)
[OK] PHP extension "mbstring" installed - required
[OK] PHP extension "openssl" installed - required
[OK] PHP extension "sodium" installed - required
[OK] Composer installed
[OK] Docker installed
[OK] Docker Compose installed
[OK] npm installed
```

*Output* de la commande "symfony book :check-requirements"

Une fois cette vérification faite, nous pouvions passer à l'installation du projet. C'est Hugo L qui s'en est chargé. Il ne restait ensuite qu'à déposer le projet installé sur le dépôt GitHub, puis aux autres membres de cloner le dépôt sur leurs ordinateurs. Pour que les autres membres du groupe aient une configuration identique du projet, ils devaient procéder à des étapes supplémentaires :

- Installer Symfony avec la commande *composer install*
- Créer le fichier *.env.local*, car ce dernier n'était pas présent sur le dépôt, étant situé dans le fichier *.gitignore*

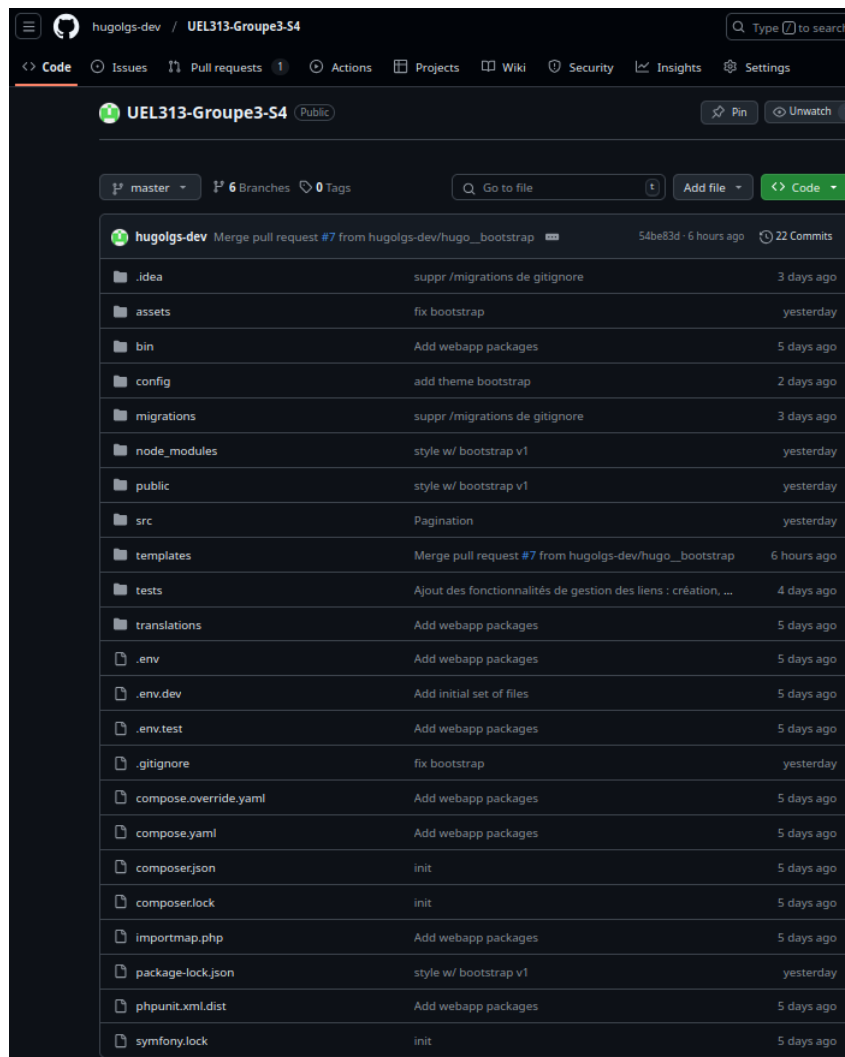
Pour l'installation, nous avons suivi à la lettre la ressource de cours concernant l'installation de Symfony 6.4 avec le CLI. Il n'y a eu aucun problème durant cette installation.

## 3.2 Dépôt GitHub du projet

URL du dépôt GitHub du projet : <https://github.com/hugolgs-dev/UEL313-Groupe3-S4>

Le dépôt GitHub est organisé de la même manière que la majorité des projets sous Symfony. Les répertoires importants sont :

- **src/** : C'est ici que se trouve la majorité du code source du projet, avec notamment les controllers et les entités ;
- **templates/** : Ce répertoire regroupe tous les fichiers Twig, responsables du rendu des pages ;
- **public/** : On y trouve les fichiers accessibles publiquement, depuis le site web : les fichiers CSS, JavaScript, ou encore les images.



Dépôt GitHub du projet

### 3.3 Création du CRUD

Dans un premier temps, nous avons mis en place l'entité *Link.php*, grâce à Doctrine [2], l'ORM de Symfony, et à la commande *php bin/console make :entity*. Ensuite, nous avons appliqué les changements grâce aux migrations, en utilisant les commandes *php bin/console make :migration* et *php bin/console doctrine :migrations :migrate*.

Une fois ces étapes passées, nous avons pu créer le CRUD, avec la commande *php bin/console make :crud*, qui permet de générer son template et son controller.

Après avoir apporté quelques finitions, le CRUD permettant de gérer des liens sur le site est entièrement opérationnel.

### 3.4 Problèmes migrations

Lors de la mise en place du CRUD, certains membres de l'équipe ont rencontré des erreurs en effectuant un *git pull* pour récupérer les modifications sur la branche *master*. Ces erreurs étaient dues à des conflits entre les migrations locales et celles présentes dans le dépôt. Par exemple, l'impossibilité de créer une nouvelle table "Links" ou un message indiquant que la "table article existe déjà".

Pour résoudre ce problème, nous avons convenu de deux points essentiels :

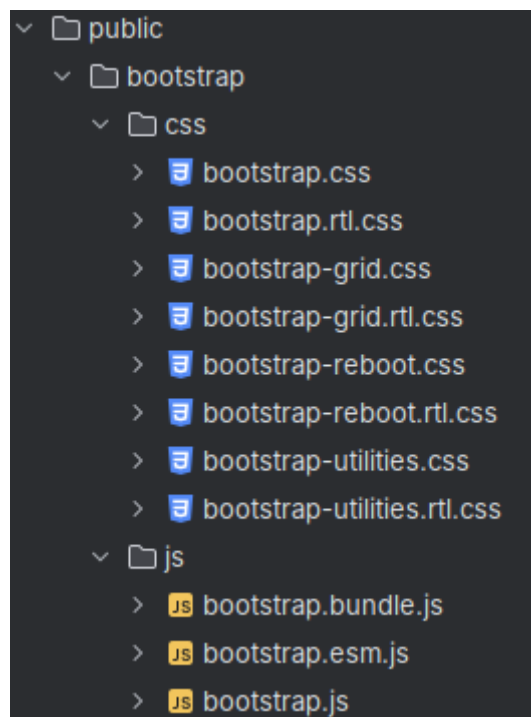
- Ajouter les fichiers de migration au fichier *.gitignore*.
- Supprimer les fichiers de migration déjà présents dans le dépôt.

En appliquant ces bonnes pratiques, nous avons évité des conflits similaires à l'avenir.

### 3.5 Ajout de Bootstrap 5.3 au projet

Dans un premier temps, nous avons installé Bootstrap via le CDN. Cependant, les changements apportés sur le HTML avec Bootstrap n'étaient pas pris en compte, bien que les fichiers s'affichaient dans l'onglet "Réseau" de l'inspecteur du navigateur.

Nous avons donc dû installer Bootstrap en téléchargeant l'archive .zip depuis le site web officiel [1]. Ensuite, nous avons placé les dossiers *css/* et *js/* dans le répertoire *public/* du site. Avec cette méthode, Bootstrap a fonctionné parfaitement.



Répertoire public du site

### 3.5.1 Utilisation de Bootstrap

Pour ce projet, notre objectif était d'obtenir un visuel simple et épuré, particulièrement au niveau des formulaires. Nous voulions que l'expérience utilisateur soit la plus agréable possible.

## Modifier un lien

### Créer ou modifier un lien

**Titre**

**URL**

**Description**

"Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in"

Modifier

Retour à la liste

Supprimer

Formulaire épuré avec Bootstrap



## 4 Conclusion

Durant cette dernière semaine, nous avons appris à collaborer efficacement et à surmonter les défis techniques ensemble. Le choix de Symfony et l'organisation mise en place nous ont permis de mener à bien le projet tout en respectant les objectifs initiaux.

## Références

- [1] *Bootstrap Download*. URL : <https://getbootstrap.com/docs/5.3/getting-started/download/>.
- [2] *Doctrine ORM*. URL : <https://symfony.com/doc/current/doctrine.html>.
- [3] *Environnement adapté à Symfony*. URL : <https://symfony.com/doc/6.4/the-fast-track/en/1-tools.html#symfony-cli>.
- [4] *Symfony CLI*. URL : <https://github.com/symfony-cli/symfony-cli>.
- [5] *Symfony Documentation*. URL : <https://symfony.com/doc/6.4/index.html>.