Aix Marseille Université | IUT

Département Informatique - Aix

Rapport

Choix techniques et démarche de maintenance

S6

Présenté(e)s par :

Félicité GARY

Yousra MEHDI

SOMMAIRE

➔Choix du framework backend

Symfony 7.4 LTS

Nous avons choisi d'utiliser **Symfony 7.4 LTS** comme framework backend pour ce projet.

Ce choix repose sur plusieurs arguments techniques et méthodologiques :

- **Version LTS (Long Term Support)** : Symfony 7.4 bénéficie d'un support étendu, avec des correctifs de bugs jusqu'en **novembre 2028** et des mises à jour de sécurité jusqu'en **novembre 2029**.

- **Stabilité et maturité** : Symfony 7.4 est une version stable, issue de la continuité des versions 7.x, sans rupture fonctionnelle majeure. Elle ne constitue pas une refonte, mais une évolution nettoyée des versions précédentes.

- **Maintenance à long terme** : contrairement à Symfony 6.4, dont le support de bugs s'arrête en 2026, Symfony 7.4 est plus adaptée à un projet développé aujourd'hui et amené à évoluer.Symfony de plus fournit nativement de nombreux mécanismes de sécurité intégrés tel que la gestion des rôles et des permissions,une protection CSRF, un hachage sécurisé des mots de passe ou encore la validation des entrées utilisateur.

- **KISS :** grâce à l'architecture donnée on peut suivre le concepte KISS (front-end twig css basique , back-end structuré)

- **Compatibilité technique** : Symfony 7.4 est pleinement compatible avec PHP 8.2 et 8.3, ainsi qu'avec les principaux composants de l'écosystème (Doctrine, Twig, Security, Validator, PHPUnit Bridge).

- **Symfony est par ailleurs un framework largement adopté** dans le monde professionnel (BlaBlaCar, Dailymotion, Decathlon, Spotify en partie), disposant d'un écosystème riche et d'une communauté active.

➔ Qualité du code et maintenance

Linting et conventions – PHP-CS-Fixer

Le projet utilise **PHP-CS-Fixer** pour garantir le respect des conventions de codage.Cet outil a été choisi car :

- standard dans l'écosystème PHP,
- largement utilisé avec Symfony,
- simple à configurer,
- reconnu par la communauté.

Analyse statique – PHPStan (niveau 6)

Nous avons pensé à ajouter une analyse statique qui est mise en place via **PHPStan**, configuré au **niveau 6**.

Le niveau 6 a été retenu car il est **exigeant mais réaliste** pour un projet de niveau BUT selon les recommandations trouvé sur internets(IA).Celui-ci détecte les erreurs de typage, incohérences logiques et usages risqués, il améliore donc fortement la fiabilité du code sans être excessivement bloquant.

Exemples de vérifications effectuées :

- appels de méthodes inexistantes,
- types de retour incohérents,
- paramètres incorrects,
- accès à des propriétés non définies.

Métriques de maintenabilité – PHP Metrics

Afin d'aller au-delà des simples conventions, le projet intègre une analyse des **métriques de maintenance** via **PHP Metrics**.

Cet outil permet de calculer :

- la complexité cyclomatique,
- l'indice de maintenabilité (Maintainability Index),
- la taille et la complexité des classes et méthodes.

L'indice de maintenabilité est un score compris entre 0 et 100 :

- ≥ 85 : très bonne maintenabilité
- 65 – 84 : bonne maintenabilité
- < 65 : à améliorer

L’objectif du projet est de maintenir un indice **supérieur à 65**, garantissant un code lisible, testable et évolutif.

➔Automatisation et prévention des erreurs

Hook Git pre-commit

Un **hook Git pre-commit** est mis en place afin d’empêcher l’envoi de code non conforme.

Avant chaque commit, les vérifications suivantes sont exécutées automatiquement :

- PHP-CS-Fixer (dry-run),
- PHPStan,
- PHPUnit.

Cela permet de :

- détecter les erreurs le plus tôt possible,
- éviter l’introduction de code cassé dans le dépôt,
- responsabiliser chaque développeur sur la qualité du code.

Intégration continue – GitHub Actions

Une **GitHub Action CI** est configurée pour s’exécuter à chaque push et pull request.Elle reproduit les mêmes vérifications que le hook local :

- installation des dépendances,
- linting,
- analyse statique,
- tests unitaires.

Cette double vérification (locale + serveur) garantit que :

- le code fonctionne sur un environnement propre,
- aucune erreur ne passe lors d’un merge.

➔Conclusion

L’utilisation de Symfony 7.4 LTS, combinée à des outils de linting, d’analyse statique, de métriques et d’automatisation,nous permettent de construire une base solide de

développement qui assure un code propre et le maintien de celui-ci le plus longtemps possible.