

# **Veille Technologique – Performance & Optimisation**

## **(Symfony)**

L'objectif de cette section est d'explorer les leviers de performance et d'optimisation possibles avec Symfony. Il ne s'agit pas forcément de tout implémenter, mais de démontrer une compréhension des solutions existantes dans une démarche d'éco-conception, de sécurité et d'efficacité technique.

### **Actions possibles**

1. **Cache HTTP** (Symfony HttpCache ou API Platform cache)  
→ éviter de recalculer les réponses pour les mêmes requêtes.
2. **Cache Doctrine** (ex: Redis)  
→ limiter les appels à la base de données et améliorer la réactivité
3. **Indexation DB**  
→ vérifier que les colonnes critiques (email, list\_id, etc.) sont indexées pour accélérer les recherches.
4. **Profiler Symfony**  
→ analyser les requêtes SQL et le temps de réponse grâce à la barre de debug et au profiler intégré.
5. **Optimisation Docker**  
→ réduire la taille des images (multistage build) et améliorer la vitesse de déploiement.

Ces optimisations permettent d'améliorer la rapidité, de réduire la consommation de ressources et de renforcer la sécurité globale du projet. Elles s'inscrivent dans une logique d'éco-conception et garantissent une meilleure expérience utilisateur tout en assurant la pérennité de l'application.