

BTS SIO — Option SLAM

INITIALISATION DU FRAMEWORK LARAVEL

Mise en place de l'environnement de développement et mapping de la base GSB

Contexte Étude	Laboratoire GSB — Application AppliFrais
Période	Janvier 2026
Technologies	Laravel / Composer / PHP Eloquent ORM / Node.js / Laragon
Candidat	COUR Cyrille

SOMMAIRE

1. Objectifs du Projet et Initialisation de l'Architecture (AppliFrais)	3
2. Configuration d'Environnement et Persistance des Données SQL	3
3. Création et Configuration Évolutive des Modèles Eloquent ORM	4
4. Automatisation du Workflow Local par Scripts Windows Batch	4
5. Ouverture Technologique : Veille Sectorielle sur l'Outil Laragon	5

1. Objectifs du Projet et Initialisation de l'Architecture (AppliFrais)

Dans le cadre du contexte d'entraînement national GSB (Galaxy Swiss Bourdin), j'ai entrepris le développement de l'application AppliFrais, un outil web complet destiné à automatiser le suivi et le remboursement des fiches de frais des visiteurs médicaux. Pour ce projet d'envergure, le choix technologique s'est porté sur le framework PHP MVC de référence : Laravel.

La première étape a consisté à initialiser le squelette applicatif à la main afin de maîtriser l'arbre des dépendances du framework :

- Installation manuelle des prérequis système locaux : le serveur XAMPP, le gestionnaire de paquets PHP Composer, et l'environnement d'exécution Node.js.
- Génération native de l'arborescence via la CLI de Composer : `composer create-project laravel/laravel applifrais`
- Vérification et exécution du serveur interne de développement : `php artisan serve`

2. Configuration d'Environnement et Persistance des Données SQL

Pour lier l'application à la base de données GSB fournie, une configuration fine a été opérée sur le fichier d'environnement global `.env` à la racine du projet. Ce fichier masque les clés d'accès tout en orchestrant la couche de persistance.

Après avoir créé un conteneur vide nommé `gsb_applifrais` sur phpMyAdmin, j'ai importé le schéma SQL et adapté les directives de connexion du framework :

- `DB_CONNECTION=mysql`
- `DB_HOST=127.0.0.1`
- `DB_PORT=3306`
- `DB_DATABASE=gsb_applifrais`

3. Création et Configuration Évolutive des Modèles Eloquent ORM

Laravel s'appuie sur le pattern Active Record via son ORM intégré : Eloquent. Contrairement à une approche classique en requêtes SQL brutes, chaque table de la base de données GSB est représentée par une classe PHP (un Modèle).

J'ai instancié les entités métier nécessaires à l'aide de l'outil en ligne de commande Artisan :

- `php artisan make:model FicheFrais`
- `php artisan make:model FraisForfait`
- `php artisan make:model LigneFraisForfait`

La base de données GSB n'adoptant pas la nomenclature native de Laravel (qui attend par défaut des clés primaires nommées 'id' et des colonnes automatiques de suivi temporel 'created_at' et 'updated_at'), j'ai configuré explicitement chaque modèle PHP :

❑ **Exemple d'adaptation structurelle (Modèle FicheFrais)** : `protected $table = 'fiche_frais'; public $timestamps = false; protected $fillable = ['idVisiteur', 'mois', 'nbJustificatifs', 'montantValide', 'dateModif', 'idEtat'];`

4. Automatisation du Workflow Local par Scripts Windows Batch

Afin d'optimiser ma productivité quotidienne et de simplifier le démarrage simultané de la pile applicative, j'ai développé un script d'automatisation Windows Batch nommé `start_Applifrais.bat` placé à la racine de mon espace de travail :

- Lancement en tâche de fond du serveur de développement PHP : `php artisan serve`
- Exécution en simultané du compilateur et watcher d'assets Front-end : `npm run dev`

Cette automatisation réduit les manipulations d'ouverture de session de codage à une seule action : exécuter le script batch, après s'être assuré du démarrage initial des services Apache et MySQL via l'interface graphique de XAMPP.

5. Ouverture Technologique : Veille Sectorielle sur l'Outil Laragon

La réalisation complète de cette installation en mode natif s'est avérée indispensable pour comprendre l'écosystème global d'un projet web moderne et la gestion fine des environnements.

En parallèle, mes recherches régulières dans le cadre de ma veille technologique m'ont permis de découvrir un outil d'infrastructure locale de pointe pour l'écosystème PHP : Laragon.

Laragon se distingue par des fonctionnalités avancées d'automatisation et d'orchestration :

- Permet d'initier un projet Laravel entièrement configuré (base de données, hôte virtuel local personnalisé du type 'applifrais.test') en un seul clic.
- Gère nativement le changement de version de PHP à la volée, l'isolation des environnements et intègre des outils modernes de manière transparente.

□ **Bilan de démarche :** Bien que la découverte de Laragon m'offre une perspective d'industrialisation et un gain de temps considérable pour mes futurs déploiements, avoir effectué cette première installation manuellement (via Composer, XAMPP et la configuration pas à pas des fichiers d'environnement) m'a apporté une compréhension fine de l'architecture système de Laravel, un atout clé pour mon profil de développeur.

