

# Rapport de Stage — BTS SIO SLAM

Option SLAM · 1ère année · Semaine 5

Hidaoui Mohamed Amine — Planisphère Info

## Semaine 5 — V6 de l'application Karlia

### Contexte

La semaine 5 poursuit la mission n°2 : l'application interne de gestion des tickets Karlia. La version 5 était fonctionnelle mais conservait deux limites : une latence au chargement (données stockées en fichiers JSON) et un affichage incomplet des tickets, dû à un écart entre la structure réelle de l'API et celle attendue par l'interface. La version 6 visait à corriger ces deux points et à ajouter la mise à jour des tickets en temps réel.

### Migration du cache en base de données

Pour supprimer la latence, les données ne sont plus rappelées à l'API à chaque consultation : elles sont synchronisées une fois et stockées en base MySQL (la même que celle des comptes utilisateurs). Au rechargement ou à la reconnexion, l'affichage est instantané puisque tout est lu localement, sans appel API. Plusieurs tables de cache ont été créées : tickets\_cache (miroir des tickets), meta\_cache (statuts, priorités, pipelines), companies\_seen (lien identifiant → nom d'entreprise), contacts\_cache et sync\_status (état des synchronisations). Le script refresh.php récupère tous les tickets de façon paginée et alimente la base, avec un verrou empêchant les exécutions simultanées.

### Correction de l'affichage

Une analyse des réponses réelles de l'API a montré que l'interface avait été codée contre une structure supposée : les tickets renvoient des champs plats (id\_status, id\_customer\_supplier, creation\_date) et non des objets imbriqués, le nom d'entreprise se trouve dans le champ title, et certains endpoints (priorités, pipelines) ne sont pas exposés comme prévu. Le code a été aligné sur la structure réelle, corrigeant les anciens symptômes : statuts vides, noms d'entreprises manquants et tirets dans les colonnes.

### Mise à jour en temps réel

Les nouveaux tickets sont signalés sans recharger la page grâce à la technologie Server-Sent Events : le navigateur ouvre une connexion unique (sse.php) que le serveur alimente dès qu'un changement survient. Une seule connexion interroge périodiquement l'API pour détecter les nouveautés, met à jour la base et un signal léger ; les autres se contentent de lire ce signal. Le système distingue les nouveaux tickets, les suppressions et les modifications.

### Fonctionnalités ajoutées

- Pièces jointes locales
- Résumés automatiques par IA : AnythingLLM locale, avec repli sur l'API Claude.
- Suivi des temps par ticket, état de facturation modifiable, et notes internes.
- Refonte du thème en une interface claire et lisible, l'ancien étant trop sombre.

### Bilan

La version 6 résout les deux limites de la version précédente — la latence par la migration en base, l'affichage par l'alignement sur l'API réelle — et ajoute des fonctionnalités métier concrètes. Elle a permis de mettre en pratique la conception d'un cache en base de données, les communications en temps réel (Server-Sent Events), l'intégration d'un service d'IA et la gestion sécurisée de fichiers. La mission se poursuit.