

Learning Log (LLog) - Projet Crime Analytics

Ce document recense les défis techniques rencontrés tout au long du développement du projet "Crime Analytics Dashboard" et les solutions concrètes appliquées. Il sert de base de connaissances pour la maintenance future.

1. Architecture & Réseau Docker

Problème : "Connection Refused" entre conteneurs

- **Symptôme :** L'API Node.js échouait à se connecter à MongoDB avec l'erreur MongooseServerSelectionError: connect ECONNREFUSED 127.0.0.1:27017.
- **Cause réelle :** Dans un environnement Docker, localhost (ou 127.0.0.1) fait référence au conteneur lui-même, pas à la machine hôte ou aux autres conteneurs.
- **Solution :** Utilisation des **noms de services** définis dans docker-compose.yml comme hostnames.
 - *Avant* : mongodb://localhost:27017
 - *Après* : mongodb://police_mongo:27017 (Docker résout ce nom vers l'IP interne du conteneur).

Problème : Conflit de Ports (EADDRINUSE)

- **Symptôme :** Le conteneur police_server redémarrait en boucle.
- **Cause réelle :** Le port 5000 était déjà utilisé par un service système Windows ou une autre instance de serveur.
- **Solution :** Mappage de port explicite dans Docker Compose vers un port libre hôte (5001) tout en gardant le port interne (5000).
 - *Config* : ports: - "5001:5000"

2. Intégration Neo4j & Backend

Problème : Format des entiers Neo4j (Integer Objects)

- **Symptôme :** Le frontend React plantait ou affichait des erreurs de sérialisation JSON lors de la réception des données du graphe.
- **Cause réelle :** Le driver JavaScript de Neo4j utilise des objets Integer propriétaires pour gérer les nombres 64-bits (BigInt), qui ne sont pas supportés nativement par JSON.stringify(). Les IDs renvoyés ressemblaient à { low: 123, high: 0 }.

- **Solution :** Implémentation d'une fonction utilitaire `getNeo4jId` ou `.toNumber()` dans le backend pour convertir systématiquement ces objets en nombres natifs JavaScript avant l'envoi au client.

Problème : Orientation du graphe pour PageRank

- **Symptôme :** L'algorithme PageRank donnait un score élevé aux "soldats" et faible au "Parrain".
- **Cause réelle :** Dans notre modélisation, la relation était `(Chef)-[:DIRIGE]->(Soldat)`. PageRank calcule l'importance basée sur les liens *entrants*. Ici, les soldats recevaient les liens, donc ils semblaient "importants".
- **Solution :** Utilisation de la projection GDS avec orientation: 'REVERSE' pour inverser virtuellement le sens des flèches lors du calcul, redonnant l'importance à la source (le Chef).

3. Frontend & Visualisation (React)

Problème : Erreur CORS (Cross-Origin Resource Sharing)

- **Symptôme :** Le navigateur bloquait les requêtes Fetch du frontend (port 3000) vers l'API (port 5001).
- **Cause réelle :** Par sécurité, les navigateurs interdisent les requêtes entre origines (ports) différentes sans autorisation explicite du serveur.
- **Solution :** Installation et configuration du middleware cors sur le serveur Express : `app.use(cors());`.

Problème : Écran blanc "Graph rendering"

- **Symptôme :** Le composant ForceGraph2D causait un crash si les données n'étaient pas prêtes.
- **Cause réelle :** Tentative de rendu du graphe alors que les tableaux `nodes` et `links` étaient encore vides ou `undefined` (asynchronisme de l'appel API).
- **Solution :** Rendu conditionnel dans React. Le graphe ne s'affiche que si `graphData.nodes.length > 0`.

4. Cohérence des Données (Polyglotte)

Problème : Désynchronisation des Entités

- **Symptôme :** Un suspect créé dans Neo4j (Relation) mais absent de MongoDB (Dossier) provoquait des erreurs 404 dans l'interface "Dossier Suspect".

- **Cause réelle** : L'absence de transaction distribuée (ACID) entre deux technologies de bases de données différentes. Une insertion peut réussir dans l'une et échouer dans l'autre.
- **Solution (Mitigation)** :
 1. Création de scripts d'initialisation (init_mongo.js, init_mysql.sql) strictement alignés sur les mêmes identifiants.
 2. Dans le code de production (futur), implémentation du pattern "Saga" : si l'écriture Mongo échoue, annuler l'écriture Neo4j.