**📋 SYNTHÈSE DISCUSSION - CHUNKING JAVASCRIPT PNGDI**

**Date:** 1er juillet 2025  
**Objectif:** Résoudre l'erreur 419 CSRF pour import de gros volumes d'adhérents (10,000+)  
**Statut:** En cours de résolution - Problème d'authentification identifié

**🎯 PROBLÈME INITIAL**

* **Erreur 500** : Timeout de 30 secondes dépassé lors d'import de 10,000 adhérents
* **Cause racine** : Traitement en une seule fois provoque timeout serveur
* **Solution retenue** : Système de chunking JavaScript

**✅ RÉALISATIONS ACCOMPLIES**

**1. SYSTÈME CHUNKING JAVASCRIPT COMPLET**

**Fichier:** public/js/chunking-import.js (Version 1.1 Production)

**Classes implémentées:**

* ChunkManager : Division en lots de 100 adhérents
* ProgressTracker : Interface temps réel avec modal Bootstrap 5
* ImportProcessor : Traitement séquentiel + gestion CSRF

**Fonctionnalités:**

* ✅ Auto-détection : Chunking si >50 adhérents, sinon traitement normal
* ✅ Modal non-interruptible avec barre de progression animée
* ✅ Statistiques temps réel (traités/erreurs/temps restant)
* ✅ Boutons pause/reprendre/annuler
* ✅ 3 tentatives par chunk avec retry automatique
* ✅ Refresh CSRF automatique entre chunks
* ✅ Hook automatique sur handleAdherentFileImport()

**2. CONTRÔLEUR LARAVEL BACKEND**

**Fichier:** app/Http/Controllers/Api/ChunkProcessorController.php

**Méthodes principales:**

* processChunk() : Traitement d'un chunk de 100 adhérents max
* refreshCSRF() : Renouvellement token CSRF
* validateAndCleanAdherent() : Validation + nettoyage données
* saveAdherent() : Sauvegarde en base avec gestion anomalies

**Optimisations:**

* ✅ Limite 25 secondes pour éviter timeout
* ✅ Traitement par batch de 50 pour optimiser mémoire
* ✅ Logs détaillés pour debugging
* ✅ Gestion des erreurs avec retry intelligent

**3. ROUTES API COMPLÈTES**

**Fichier:** routes/api.php (Mise à jour)

**Routes principales:**

* POST /api/organisations/process-chunk : Traitement chunks
* GET /api/csrf-refresh : Refresh CSRF
* GET /api/chunking/health : Health check système

**Fichier:** routes/web.php (Routes alternatives)

* POST /chunking/process-chunk : Alternative route web
* GET /chunking/csrf-refresh : Alternative CSRF web

**⚠️ PROBLÈME ACTUEL - AUTHENTIFICATION**

**Symptôme**

16:20:24 - 📦 Lot 1: traitement de 100 adhérents...

16:20:24 - ❌ Lot 1: Erreur tentative 1 - Unauthenticated.

**Diagnostic en cours**

* ✅ Chunking fonctionne : Système divise et traite les chunks
* ✅ Communication serveur : Les requêtes arrivent au contrôleur
* ❌ Authentification : Erreur "Unauthenticated"

**Solutions testées**

1. **Middleware correction** : auth:sanctum → web + auth
2. **Headers CSRF** : Ajout credentials: 'same-origin'
3. **Routes alternatives** : Endpoints web dédiés
4. **Debug complet** : Logs détaillés de toutes les requêtes

**Auto-détection implémentée**

Le JavaScript teste automatiquement :

1. Routes API (/api/organisations/process-chunk)
2. Si échec → Routes Web (/chunking/process-chunk)
3. Logs détaillés pour identifier le problème exact

**📁 FICHIERS MODIFIÉS**

**1. JavaScript Principal**

// public/js/chunking-import.js - Version 1.1 Production

- Classes complètes : ChunkManager, ProgressTracker, ImportProcessor

- Auto-détection routes API/Web

- Debug complet avec logs détaillés

- Gestion CSRF robuste

**2. Contrôleur Laravel**

// app/Http/Controllers/Api/ChunkProcessorController.php

- processChunk() : Traitement optimisé par batch

- Validation complète des données

- Gestion mémoire et timeout

**3. Routes**

// routes/api.php - Routes API avec middleware web

// routes/web.php - Routes alternatives ajoutées à la fin

**🔧 CONFIGURATION ACTUELLE**

**Paramètres Chunking**

* **Taille chunk** : 100 adhérents par lot
* **Seuil déclenchement** : 50+ adhérents
* **Pause entre chunks** : 500ms
* **Tentatives max** : 3 par chunk
* **Timeout serveur** : 25 secondes (< 30s Laravel)

**Endpoints**

* **API Principal** : /api/organisations/process-chunk
* **Web Alternative** : /chunking/process-chunk
* **CSRF Refresh** : /api/csrf-refresh + /chunking/csrf-refresh
* **Health Check** : /chunking/health

**📊 TESTS EFFECTUÉS**

**JavaScript**

✅ Chunking activé : 100 chunks de 100 éléments créés  
✅ Modal affichée : Interface progression temps réel  
✅ Communication serveur : Requêtes envoyées correctement  
❌ Authentification : Erreur "Unauthenticated" persiste

**Debug en cours**

* Headers de requête examinés
* Cookies de session vérifiés
* Middleware testés (sanctum → web)
* Routes alternatives implémentées

**🚀 PROCHAINES ÉTAPES**

**Diagnostic à approfondir**

1. **Vérifier authentification session** Laravel exacte
2. **Tester routes web alternatives** après correction syntaxe
3. **Analyser logs Laravel** pour erreurs middleware
4. **Valider configuration CSRF** complète

**Solution finale attendue**

Une fois l'authentification résolue :

* ✅ Import 10,000 adhérents sans timeout
* ✅ Progression temps réel dans modal
* ✅ Gestion automatique des erreurs
* ✅ Retry intelligent avec refresh CSRF

**📋 COMMANDES DE TEST**

**Health Check**

GET http://votre-site.com/chunking/health

**Import test**

1. Fichier CSV avec 100+ adhérents
2. Observer console : tentatives API → Web
3. Vérifier progression modal

**Debug console**

// Logs attendus après correction :

🔄 Tentative 1: Route API - /api/organisations/process-chunk

❌ Échec Route API: Unauthenticated

🔄 Tentative 2: Route Web - /chunking/process-chunk

✅ Succès avec Route Web

**📌 Point de départ pour prochaine discussion :** Résoudre définitivement l'erreur "Unauthenticated" en identifiant la configuration d'authentification exacte de Laravel utilisée dans le projet PNGDI.