

CORRECTIONS PHASE 2 - P1 : Demandes de payout visibles côté admin

Date : 27 décembre 2025

Commit : À définir

Branche : feature/stripe-payout-automation

🎯 Objectif

Corriger le problème des demandes de payout invisibles côté admin et améliorer le système de notifications.

📋 Problèmes identifiés

Symptômes

- ✓ Créateur fait une demande de payout → Demande créée en base avec statut PENDING_APPROVAL
- ✗ Admin ne voit rien dans son dashboard (/dashboard/admin/payouts)
- ✗ Workflow de validation ne fonctionne pas

Causes

- Changement initial défaillant** : Le useEffect dans le composant admin ne se déclencheait pas au montage initial
- Notifications incomplètes** :
 - Fonction `createNotification()` non importée (erreur runtime)
 - Pas d'envoi d'email aux admins
- Requête API sans logs** : Impossible de déboguer les problèmes de récupération

✓ CORRECTION 1 : Chargement initial des payouts

Fichier : `app/dashboard/admin/payouts/page.tsx`

Problème

Le useEffect dépendait uniquement des `filters`, donc il ne se déclencheait pas lors du montage initial du composant.

Solution

- ✓ Ajout d'un useEffect séparé pour le montage initial (dépendances vides `[]`)
- ✓ Conservation du useEffect existant pour les changements de filtres
- ✓ Ajout de logs de débogage dans `fetchPayouts()`

Code modifié

Avant :

```
useEffect(() => {
  fetchPayouts();
}, [filters]);
```

Après :

```
// Initial load on component mount
useEffect(() => {
  console.log('[AdminPayouts] Component mounted, fetching payouts...');
  fetchPayouts();
}, []); // Empty dependencies = runs only on mount

// Reload when filters change
useEffect(() => {
  console.log('[AdminPayouts] Filters changed:', filters);
  fetchPayouts();
}, [filters]);
```

Logs ajoutés dans `fetchPayouts()`

```
console.log('[AdminPayouts] Fetching payouts with params:', params.toString());
console.log('[AdminPayouts] Response:', { ok: response.ok, status: response.status, count: data.length });
console.log('[AdminPayouts] Payouts loaded:', data.length);
```

✓ CORRECTION 2 : Notifications admin robustes

Fichier : `app/api/payouts/request/route.ts`

Problèmes

1. Fonction `createNotification()` appelée mais non importée
2. Pas d'envoi d'email aux admins
3. Pas de gestion d'erreurs pour les notifications

Solution

1. ✓ Import des fonctions nécessaires
2. ✓ Création de notifications in-app robustes
3. ✓ Ajout d'envoi d'email HTML professionnel aux admins
4. ✓ Gestion des erreurs (n'empêche pas la création du payout)
5. ✓ Récupération de la relation `creator.user` pour afficher le nom

Imports ajoutés

```
import { createNotification } from '@/lib/notifications';
import { sendEmail } from '@/lib/email';
```

Modification de la requête creator

```
const creator = await prisma.creator.findUnique({
  where: { id: creatorId },
  include: {
    user: {
      select: {
        id: true,
        name: true,
        email: true,
      },
    },
  },
});
```

Système de notification complet

Notifications in-app :

```
for (const admin of admins) {
  try {
    await createNotification({
      userId: admin.id,
      type: 'SYSTEM',
      title: '⚠ Nouvelle demande de paiement',
      message: `${creator.user.name} a demandé un paiement de ${payoutAmount.toFixed(2)} EUR. Veuillez approuver ou rejeter la demande.`,
      link: '/dashboard/admin/payouts',
    });
    console.log(`[Payout] In-app notification sent to admin ${admin.id}`);
  } catch (notifError) {
    console.error(`[Payout] Error creating in-app notification for admin ${admin.id}:`, notifError);
  }
}
```

Emails HTML professionnels :

- Template HTML responsive avec gradient Call a Star
- Affichage des détails de la demande (créateur, montant, devise, conversion si applicable)
- Bouton CTA “Gérer la demande” vers /dashboard/admin/payouts
- Gestion des erreurs d’envoi (n’empêche pas la création du payout)

Logs ajoutés

```
console.log('[Payout] Notifying admins of new payout request...');
console.log(`[Payout] Found ${admins.length} admin(s) to notify`);
console.log(`[Payout] In-app notification sent to admin ${admin.id}`);
console.log(`[Payout] Email notification sent to admin ${admin.email}`);
console.log(`[Payout] Admin notifications completed`);
```

CORRECTION 3 : Requête API améliorée

Fichier : `app/api/admin/payouts/route.ts`

Problèmes

1. Pas de logs pour déboguer
2. Where clause difficile à tracer
3. Pas de statistiques sur les résultats

Solution

1.  Ajout de logs détaillés à chaque étape
2.  Construction explicite du where clause
3.  Logs des statistiques par statut
4.  Ajout du champ `id` dans la sélection user

Code modifié

Where clause amélioré :

```
const whereClause: any = {};

if (status && status !== 'all' && status !== '') {
  whereClause.status = status;
  console.log('[AdminPayouts] Filtering by status:', status);
}

if (creatorId && creatorId !== 'all' && creatorId !== '') {
  whereClause.creatorId = creatorId;
  console.log('[AdminPayouts] Filtering by creatorId:', creatorId);
}

console.log('[AdminPayouts] Where clause:', JSON.stringify(whereClause));
```

Statistiques après récupération :

```
console.log(`[AdminPayouts] Found ${payouts.length} payout(s)`);

const statusCounts = payouts.reduce((acc, p) => {
  acc[p.status] = (acc[p.status] || 0) + 1;
  return acc;
}, {} as Record<string, number>);
console.log('[AdminPayouts] Payouts by status:', statusCounts);
```

Tests à effectuer

Test 1 : Chargement initial

1.  Se connecter en tant qu'admin
2.  Accéder à `/dashboard/admin/payouts`
3.  Vérifier que les payouts se chargent automatiquement
4.  Vérifier les logs dans la console : `[AdminPayouts] Component mounted, fetching payouts...`

Test 2 : Création de demande de payout

1. Se connecter en tant que créateur
2. Créer une demande de payout
3. Vérifier les logs : [Payout] Notifying admins of new payout request...
4. Vérifier que la demande est créée avec statut PENDING_APPROVAL

Test 3 : Notifications admin

1. Vérifier la notification in-app dans le dashboard admin
2. Vérifier la réception d'email par l'admin
3. Vérifier le contenu de l'email (détails, bouton CTA)

Test 4 : Visibilité côté admin

1. Rafraîchir le dashboard admin /dashboard/admin/payouts
2. Vérifier que la nouvelle demande apparaît
3. Vérifier le statut PENDING_APPROVAL
4. Vérifier les boutons "Approuver" et "Rejeter"

Test 5 : Filtres

1. Filtrer par statut PENDING_APPROVAL
 2. Vérifier que les filtres fonctionnent
 3. Vérifier les logs : [AdminPayments] Filters changed: { status: 'PENDING_APPROVAL' }
-



Impact attendu

Avant les corrections

- Dashboard admin vide (payouts invisibles)
- Admin non notifié des nouvelles demandes
- Workflow de validation cassé
- Impossible de déboguer

Après les corrections

- Dashboard admin affiche tous les payouts au chargement
 - Admin reçoit notification in-app + email pour chaque demande
 - Workflow de validation fonctionnel
 - Logs détaillés pour le débogage
 - Filtres fonctionnels
-

Logs de débogage

Console navigateur (Admin Dashboard)

```
[AdminPayouts] Component mounted, fetching payouts...
[AdminPayouts] Fetching payouts with params:
[AdminPayouts] Response: [ ok: true, status: 200, count: 3 ]
[AdminPayouts] Payouts loaded: 3
```

Console serveur (API)

```
[AdminPayouts] Fetching payouts with filters: [ status: null, creatorId: null ]
[AdminPayouts] Where clause: []
[AdminPayouts] Found 3 payout(s)
[AdminPayouts] Payouts by status: [ PENDING_APPROVAL: 2, PAID: 1 ]
```

Console serveur (Création demande)

```
[Payout] Currency conversion: 100 EUR -> 85.50 GBP (rate: 0.855)
[Payout] Notifying admins of new payout request...
[Payout] Found 2 admin(s) to notify
[Payout] In-app notification sent to admin user123
[Payout] Email notification sent to admin admin@callastar.fr
[Payout] In-app notification sent to admin user456
[Payout] Email notification sent to admin admin2@callastar.fr
[Payout] Admin notifications completed
```



Notes techniques

Dépendances utilisées

- @/lib/notifications : Création de notifications in-app
- @/lib/email : Envoi d'emails via Resend
- Prisma relations : creator.user pour récupérer les infos utilisateur

Gestion d'erreurs

- Les erreurs de notification n'empêchent pas la création du payout
- Logs détaillés pour chaque erreur
- Try-catch autour de chaque notification

Performance

- Chargement initial unique au montage
- Rechargement uniquement sur changement de filtres
- Requêtes optimisées avec include au lieu de requêtes séparées

Prochaines étapes

Phase 2 - P2 (Priorité moyenne)

- Améliorer les templates d'email (branding Call a Star)
- Ajouter des notifications push (optionnel)
- Dashboard des notifications admin

Phase 3 (Améliorations futures)

- Système de rappels pour les demandes en attente > 24h
- Statistiques temps de traitement des demandes
- Workflow d'approbation multi-niveaux

Fichiers modifiés

1.  app/dashboard/admin/payouts/page.tsx
 - Ajout useEffect séparé pour montage initial
 - Ajout logs de débogage
2.  app/api/payouts/request/route.ts
 - Import des fonctions de notification et email
 - Récupération relation creator.user
 - Système complet de notifications (in-app + email)
 - Logs détaillés
3.  app/api/admin/payouts/route.ts
 - Logs détaillés de la requête
 - Where clause explicite
 - Statistiques par statut
 - Ajout champ `id` dans user select

Checklist de validation

- [x] CORRECTION 1 : Chargement initial fonctionnel
- [x] CORRECTION 2 : Notifications admin (in-app + email)
- [x] CORRECTION 3 : Requête API avec logs
- [x] Code testé (compilation OK)
- [x] Documentation créée
- [] Tests manuels effectués
- [] Commit créé

Auteur : DeepAgent

Date : 27 décembre 2025

Statut :  Code implémenté, en attente de tests manuels