



Correction Agrégation Multi-Devises Dashboard Admin

Date : 27 décembre 2024

Commit : 8147cb7

Statut : Terminé



Problème Résolu

Problème initial : Les statistiques admin agrégeaient les montants sans tenir compte de la devise :

- Exemple : 1000 CHF + 500 EUR = 1500 (sans unité ❌)
- Totaux trompeurs et inexacts
- Fees calculés sans distinction de devise

Solution : Affichage des totaux séparés par devise avec calculs corrects



Modifications Effectuées

1. API Backend - Groupement par Devise

app/api/admin/payouts/dashboard/route.ts

- Agrégations groupées par `currency` avec `GROUP BY`
- Retourne des objets par devise :

```
typescript
{
  totalPayoutVolumeByCurrency: { CHF: 1000.00, EUR: 500.00 },
  totalFeesByCurrency: { CHF: 100.00, EUR: 50.00 },
  totalReadyAmountByCurrency: { CHF: 200.00, EUR: 100.00 }
}
```
- Calcul des fees par devise (10% configurable via PlatformSettings)
- **Intégration API Stripe Balance :**
- `stripe.balance.retrieve()` pour montants disponibles et en attente
- Conversion automatique centimes → unités (`amount / 100`)
- Format : `{ CHF: { available: 150.00, pending: 50.00 }, ... }`

app/api/admin/dashboard/route.ts

- Revenus et commissions groupés par devise
- Structure : `revenueByCurrency: { CHF: { totalRevenue: 1000, totalCommissions: 100 }, ... }`

2. Composant MultiCurrencyDisplay

`components/admin/MultiCurrencyDisplay.tsx` (NOUVEAU)

Composant réutilisable pour afficher des montants multi-devises :

Fonctionnalités :

- ☒ Accepte `Record<string, number>` (ex: `{ CHF: 1000, EUR: 500 }`)
- ☒ Tri alphabétique des devises pour cohérence
- ☒ Orientations : `vertical` (par défaut) ou `horizontal`
- ☒ Utilise `CurrencyDisplay` pour formatage correct de chaque devise

Variantes :

- `MultiCurrencyDisplay` : Affichage standard
- `MultiCurrencyDisplayCompact` : Inline avec séparateurs (`1,000.00 CHF | 500.00 EUR`)
- `MultiCurrencyDisplayCard` : Avec bordure et fond

Export :

```
export {
  MultiCurrencyDisplay,
  MultiCurrencyDisplayCompact,
  MultiCurrencyDisplayCard
} from '@components/admin';
```

3. Frontend Dashboard Payouts

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

Interface TypeScript mise à jour :

```
interface DashboardData {
  payoutVolume30Days: {
    totalAmountByCurrency: Record<string, number>;
    totalFeesByCurrency: Record<string, number>;
    count: number;
  };
  readyPayments: {
    totalAmountByCurrency: Record<string, number>;
  };
  stripeBalance?: Record<string, { available: number; pending: number }>;
  // ...
}
```

Affichages corrigés :

1. Carte Volume (30 derniers jours) :

- Montants par devise avec `MultiCurrencyDisplay`
- Fees plateforme affichées séparément par devise

1. Carte Paiements prêts :

- Montants par devise en mode horizontal

2. Nouvelle Carte : Balance Stripe :

- Affiche disponible et en attente par devise
- Design avec codes couleur (vert = disponible, jaune = en attente)

3. Paiements échoués :

- Affichage de la devise pour chaque payout avec `CurrencyDisplay`

4. Frontend Dashboard Principal

`app/dashboard/admin/page.tsx`

Modifications :

- ☒ Import de `MultiCurrencyDisplay`

- ☒ Carte **Revenus totaux** :

`tsx`

```
<MultiCurrencyDisplay
  amounts={Object.entries(revenueByCurrency).reduce((acc, [currency, data]) => {
    acc[currency] = data.totalRevenue;
    return acc;
  }, {})}
  orientation="vertical"
/>
```

- ☒ Carte **Commissions** : Même logique avec `totalCommissions`

Exemples d’Affichage

Avant

Total Payouts: 1500.00 € (incorrect - mélange CHF + EUR)
Total Fees: 150.00 € (incorrect)

Après ☒

Total Payouts:
- 1,000.00 CHF
- 500.00 EUR

Total Fees:
- 100.00 CHF
- 50.00 EUR

Stripe Balance:

 Disponible	 En attente
- 150.00 CHF	- 50.00 CHF
- 75.00 EUR	- 25.00 EUR

Tests Recommandés

Pour valider les corrections :

1. Créer des payouts dans différentes devises :

`bash`

```
# Via admin UI ou API
```

```
POST /api/admin/payouts
```

```
{ "creatorId": "...", "amount": 100, "currency": "CHF" }
```

```
{ "creatorId": "...", "amount": 50, "currency": "EUR" }
```

2. Vérifier Dashboard Payouts (/dashboard/admin/payouts/dashboard) :

- ☒ Volumes affichés séparément par devise
- ☒ Fees calculés correctement (10% de chaque devise)
- ☒ Stripe Balance affiché avec montants disponibles/en attente par devise

3. Vérifier Dashboard Principal (/dashboard/admin) :

- ☒ Revenus totaux affichés par devise
- ☒ Commissions affichées par devise

4. Vérifier les statistiques :

- ☒ Aucun mélange de devises
- ☒ Chaque devise affichée avec son code correct (CHF, EUR, etc.)



Devises Supportées

- **CHF** : Franc Suisse
- **EUR** : Euro
- **Extensible** : Le système s'adapte automatiquement à toute nouvelle devise ajoutée



Configuration

Commission Plateforme

La commission est configurable via l'interface admin :







- Par défaut : **10%**
- Modifiable dans : /dashboard/admin → Paramètres
- Stockée dans : PlatformSettings.platformCommissionRate

API Stripe Balance

- Activée automatiquement si STRIPE_SECRET_KEY est configurée
- En cas d'erreur Stripe, les statistiques continuent de s'afficher (balance vide)
- Gestion d'erreur robuste avec try/catch



Fichiers Modifiés

-  app/api/admin/dashboard/route.ts
-  app/api/admin/payouts/dashboard/route.ts
-  app/dashboard/admin/page.tsx
-  app/dashboard/admin/payouts/dashboard/page.tsx
-  components/admin/index.ts
-  components/admin/MultiCurrencyDisplay.tsx (NOUVEAU)

Prochaines Étapes (Optionnel)

1. **Tests E2E** : Automatiser les tests de vérification multi-devises
 2. **Graphiques** : Ajouter des graphiques de revenus par devise
 3. **Exports** : Permettre l'export CSV des statistiques par devise
 4. **Filtres** : Ajouter des filtres par devise dans les tableaux admin
-

Ressources Utiles

- **Utilitaire devises** : `/lib/currency-utils.ts`
 - **Composant CurrencyDisplay** : `/components/admin/CurrencyDisplay.tsx`
 - **Documentation Stripe Balance** : https://stripe.com/docs/api/balance/balance_retrieve
-

 **Correction validée et commitée !**