





Instructions pour Pusher les Commits sur GitHub

✓ État Actuel

Branche : `feature/stripe-payout-automation`

Commits locaux prêts à être pushés : 7 commits

Liste des Commits Locaux

```
e5ffbeb chore: Add .abacus.donotdelete to .gitignore
3e4e503  Phase 3: Documentation et tests
8677ffc  Phase 3: Créer l'entité Payout métier complète
a5a510a  Phase 2: Complete notification system implementation
59aa495 feat: Phase 1.2 - Gestion des Refunds et Disputes
5eef44d  Documentation: Phase 1.1 - Separate Charges and Transfers
76ae647 Phase 1.1: Refactoring - Separate Charges and Transfers
```

Problème Rencontré


L'authentification GitHub n'est pas configurée sur cette machine. Le push a échoué avec :

```
fatal: could not read Username for 'https://github.com': No such device or address
```

Solutions Recommandées

Option 1 : Utiliser un Personal Access Token (PAT) GitHub

Étape 1 : Créer un PAT sur GitHub

1. Aller sur GitHub : `https://github.com/settings/tokens`
2. Cliquer sur "Generate new token" → "Generate new token (classic)"
3. Donner un nom au token : `callastar-deployment`
4. Sélectionner les scopes :
 -  `repo` (accès complet aux repositories)
 -  `workflow` (pour les GitHub Actions)
5. Cliquer sur "Generate token"
6. **Copier le token immédiatement** (il ne sera plus visible après)

Étape 2 : Configurer le PAT sur la machine

```
cd /home/ubuntu/callastar

# Méthode 1 : Utiliser git credential helper (recommandé)
git config --global credential.helper store
git push origin feature/stripe-payout-automation
# Entrer votre username GitHub
# Entrer le PAT comme mot de passe

# Méthode 2 : Inclure le token dans l'URL (temporaire)
git remote set-url origin https://USERNAME:TOKEN@github.com/StreallyX/callastar.git
git push origin feature/stripe-payout-automation
# Remettre l'URL normale après le push
git remote set-url origin https://github.com/StreallyX/callastar.git
```

Option 2 : Utiliser SSH (Recommandé pour usage permanent)

Étape 1 : Générer une clé SSH

```
ssh-keygen -t ed25519 -C "votre-email@example.com"
# Accepter le chemin par défaut (~/.ssh/id_ed25519)
# Définir une passphrase (optionnel)
```

Étape 2 : Ajouter la clé SSH à GitHub

```
# Copier la clé publique
cat ~/.ssh/id_ed25519.pub

# Aller sur GitHub : https://github.com/settings/keys
# Cliquer sur "New SSH key"
# Coller la clé publique
```

Étape 3 : Changer l'URL du remote

```
cd /home/ubuntu/callastar
git remote set-url origin git@github.com:StreallyX/callastar.git
git push origin feature/stripe-payout-automation
```

Option 3 : Utiliser GitHub CLI (gh)

```
# Installer GitHub CLI
curl -fsSL https://cli.github.com/packages/githubcli-archive-keyring.gpg | sudo dd
of=/usr/share/keyrings/githubcli-archive-keyring.gpg
echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/githubcli-
archive-keyring.gpg] https://cli.github.com/packages stable main" | sudo tee /etc/apt/
sources.list.d/github-cli.list > /dev/null
sudo apt update
sudo apt install gh -y

# S'authentifier
gh auth login
# Choisir "GitHub.com"
# Choisir "HTTPS"
# Authentifier via le navigateur

# Pusher
cd /home/ubuntu/callastar
git push origin feature/stripe-payout-automation
```



Résumé des Changements à Pusher

Phase 1.1 : Separate Charges and Transfers

- **Commits :** 76ae647 , 5eef44d
- **Changements :**
 - Créateur reçoit toujours 85 EUR (pas 81.80)
 - Plateforme absorbe les frais Stripe
 - Documentation complète

Phase 1.2 : Gestion Refunds et Disputes

- **Commit :** 59aa495
- **Changements :**
 - Modèles Refund et Dispute avec tracking dette
 - Webhooks charge.refunded, charge.dispute.*
 - Transfer Reversal automatique
 - UI admin /dashboard/admin/refunds-disputes

Phase 2 : Système de Notifications

- **Commit :** a5a510a
- **Changements :**
 - Modèle Notification avec enum NotificationType
 - API routes complètes (GET, PATCH, mark-all-read)
 - Composant NotificationBell avec badge
 - Intégration dans tous les workflows

Phase 3 : Entité Payout Métier

- **Commits :** 8677ffc , 3e4e503
- **Changements :**

- Enum PayoutStatus (REQUESTED, APPROVED, PROCESSING, PAID, FAILED, REJECTED, CANCELED)
- Modèle Payout complet avec audit trail
- API routes (request, approve, reject)
- Webhooks Stripe (payout.paid, payout.failed)
- UI admin avec filtres et actions
- Documentation et tests

Chore : Gitignore

- **Commit** : e5ffbcb
- **Changements** :
- Ajout de .abacus.donotdelete au .gitignore

Liens GitHub (après push)

- **Branche** : <https://github.com/StreallyX/callastar/tree/feature/stripe-payout-automation>
- **Comparer avec main** : <https://github.com/StreallyX/callastar/compare/main...feature/stripe-payout-automation>
- **Créer une Pull Request** : <https://github.com/StreallyX/callastar/pull/new/feature/stripe-payout-automation>

Commande Finale de Push

Une fois l'authentification configurée :

```
cd /home/ubuntu/callastar
git push origin feature/stripe-payout-automation
```

Vérification Post-Push

Après le push réussi, vérifier :






```
# Vérifier que les commits sont synchronisés
git log origin/feature/stripe-payout-automation..HEAD --oneline
# Doit retourner : (rien)

# Vérifier le statut
git status
# Doit retourner : "Your branch is up to date with 'origin/feature/stripe-payout-automation'"

```

Prochaines Étapes

1.  Configurer l'authentification GitHub (choisir une option ci-dessus)

2.  Exécuter `git push origin feature/stripe-payout-automation`
3.  Créer une Pull Request vers `main`
4.  Faire une review du code
5.  Merger la PR
6.  Déployer en production