

Procédure de Récupération et de Correction après Échec des Tests

Lorsque les tests automatisés échouent dans la branche `dev`, le workflow GitHub Actions merge automatiquement le code dans la branche `failure`. Pour récupérer cette branche, apporter des corrections et les repousser sur `dev`, suivez les étapes ci-dessous :

Étape 0 : Boire un shooter



Étape 1 : Récupération de la branche `failure`

```
# Assurez-vous d'être sur votre branche locale `dev`  
git checkout dev  
  
# Récupérez les derniers changements du dépôt, y compris la branche `failure`  
git fetch origin  
  
# Fusionnez les changements de la branche `failure` dans `dev`  
git merge origin/failure
```

Étape 2 : Identification et Correction des Problèmes

- Analysez les logs d'échec des tests pour identifier le problème
- Faites les modifications nécessaires pour corriger les erreurs
- N'oubliez pas de mettre à jour ou d'ajouter des tests pour couvrir les corrections apportées

Étape 3 : Testez vos corrections localement

Exécutez vos tests localement pour vous assurer que tout passe:

```
pytest .\test\
```

Étape 4 : Commit et Push des Corrections

- Ajoutez vos modifications à l'index Git

```
git add .
```

- Committez vos modifications avec un message descriptif

```
git commit -m "Corrections des erreurs blablabla"
```

- Poussez la branche de corrections sur GitHub

```
git push origin dev
```

Étape 6 : Fusionnez fix-failure dans dev

Une fois la Pull Request approuvée, fusionnez-la dans dev. GitHub Actions exécutera automatiquement les tests à nouveau.

Étape 7 : Nettoyage

Après la fusion, vous pouvez supprimer la branche de corrections si vous le souhaitez :

```
git branch -d fix-failure  
git push origin --delete fix-failure
```

Cette procédure doit être suivie à chaque fois que les tests échouent pour garantir que les corrections sont bien intégrées et que la branche dev reste stable.