



TP : Transfert d'un Projet vers GitHub

Authentification par Jeton et Fusion de Branches

Contexte et Mission



Situation de départ :

- Le projet n'existe que sur votre machine.
- La branche `master` contient l'historique initial.
- La branche `reprise` contient la version préférée.
- Aucun dépôt GitHub n'est encore lié au projet.

Mission :

Transférer notre travail sur GitHub de manière sécurisée et faire de la version `reprise` la nouvelle version officielle sur la branche `master`.

Objectifs Pédagogiques



À l'issue de ce TP, vous serez capable de :

- **S'authentifier** sur GitHub avec un jeton d'accès personnel (token).



- **Envoyer** une branche locale vers un dépôt distant sur GitHub.



- **Fusionner** une branche dans une autre.



- **Comprendre** l'articulation entre les branches locales et distantes.

Prérequis



Assurez-vous que votre environnement est prêt :

1. **TP Git Local Terminé** : Vous maîtrisez les commandes de base.
2. **Branches Existantes** : Votre dépôt local contient les branches `master` et `reprise`.
3. **Compte GitHub** : Vous disposez d'un compte GitHub fonctionnel.

Pour vérifier vos branches locales, exécutez :

```
```bash git branch ```
```

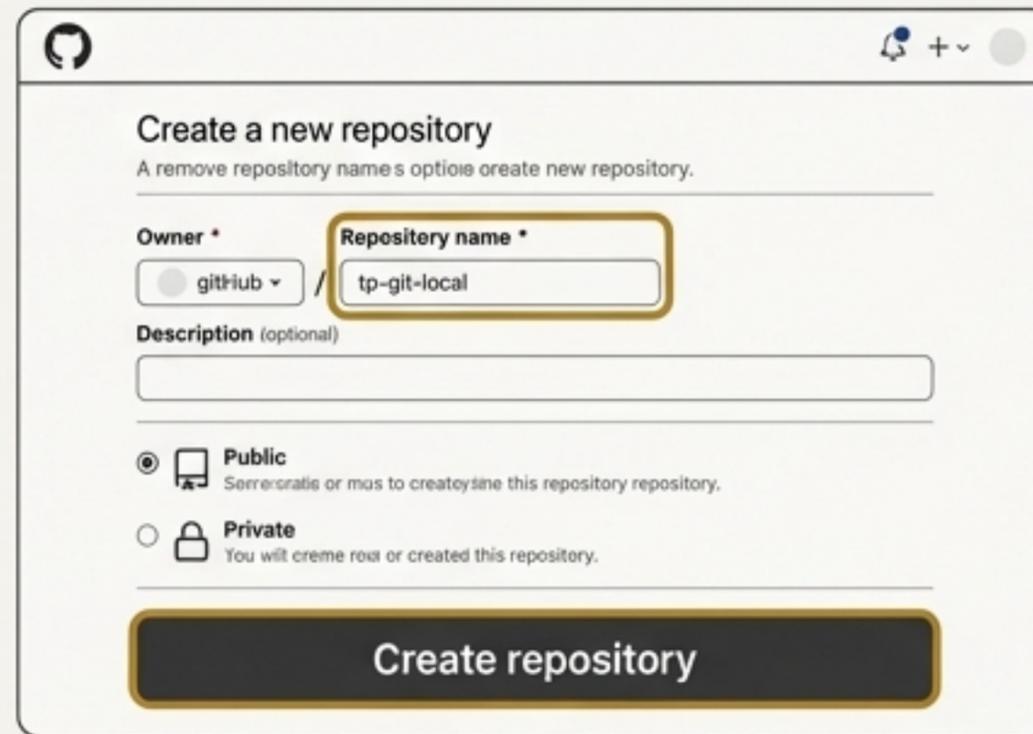
Le résultat attendu est `\* reprise` et `master`.

# Étape 1 : Lier le Local au Distant

Nous allons d'abord établir le "pont" entre notre machine et GitHub.

- **1. Créez un nouveau dépôt sur GitHub :**

- Nommez-le (ex: `tp-git-local`).
- **Important :** N'initialisez **AUCUN** fichier (pas de README, .gitignore, ou licence).
- Copiez l'URL HTTPS du dépôt.



Create a new repository  
A repository name is required to create a new repository.

Owner \*  /

Repository name \*

Description (optional)

Public  
Anyone can create or modify this repository.

Private  
Only you can create or modify this repository.

Create repository



Dépôt Local



Dépôt Distant

- **2. Liez votre dépôt local à ce dépôt distant :**

```
'origin' est le nom par défaut pour la connexion distante
git remote add origin
https://github.com/VOTRE_USER/VOTRE_DEPOT.git
```

- **3. Vérifiez la connexion :**

```
git remote -v
```

# Étape 2 : S'authentifier avec un Jeton d'Accès

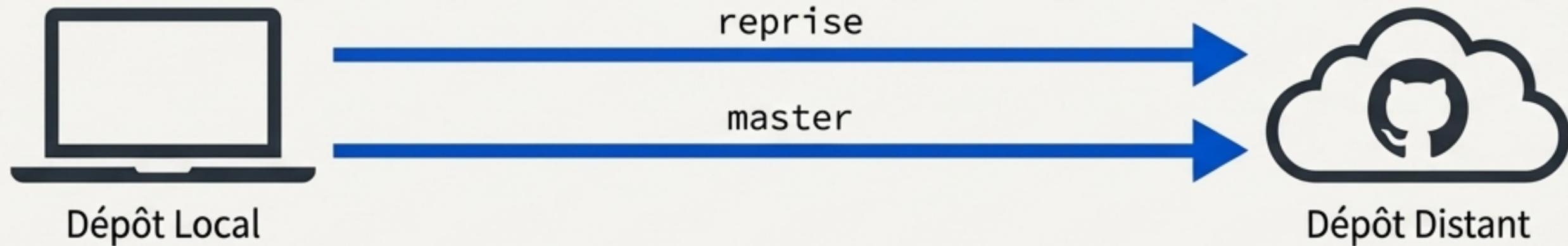


GitHub requiert un Jeton d'Accès Personnel (PAT) à la place du mot de passe pour les opérations Git. C'est votre clé sécurisée pour accéder à vos dépôts.

## Procédure de création sur GitHub :

1. Allez dans `Settings` > `Developer settings`.
2. `Personal access tokens` > `Tokens (classic)`.
3. Générez un nouveau jeton.
4. **Permissions minimales requises** : Cochez `repo`, ce qui inclut les droits `Contents: Read and write`.
5. **Copiez et conservez précieusement ce jeton. Il ne sera plus jamais affiché.**

## Étape 3 : Envoyer les Branches sur GitHub



Maintenant que le pont est établi et que nous avons la clé, nous pouvons envoyer notre travail.

1. Envoyer la branche ``reprise`` :

```
L'option -u établit un lien de suivi entre la branche locale et distante
git push -u origin reprise
```

Lorsqu'on vous le demande, entrez votre login GitHub et **collez le jeton (PAT)** en guise de mot de passe.

2. Envoyer la branche ``master`` :

```
git switch master
git push -u origin master
```

**À ce stade, vos deux branches locales sont désormais sauvegardées et visibles sur GitHub.**

# Étape 4 : Fusionner le Travail en Local

L'objectif est d'intégrer les améliorations de reprise dans notre branche principale master. Cette opération est d'abord réalisée et validée en local.

1. Se positionner sur la branche qui va recevoir la fusion :

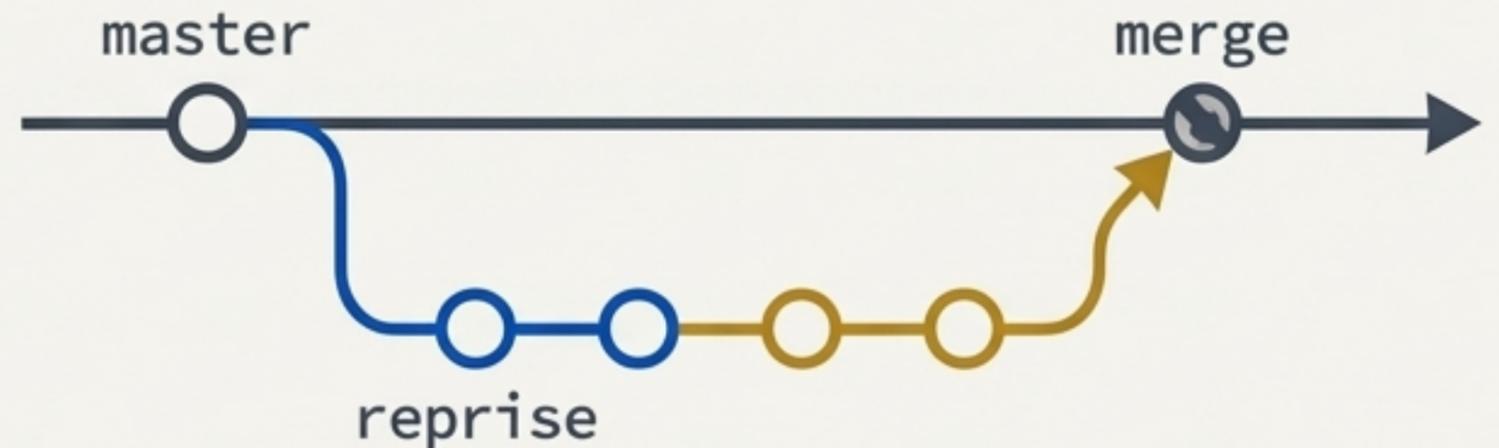
```
git switch master
```

2. Exécuter la fusion :

```
On "merge" la branche 'reprise' DANS la
branche actuelle ('master')
git merge reprise
```

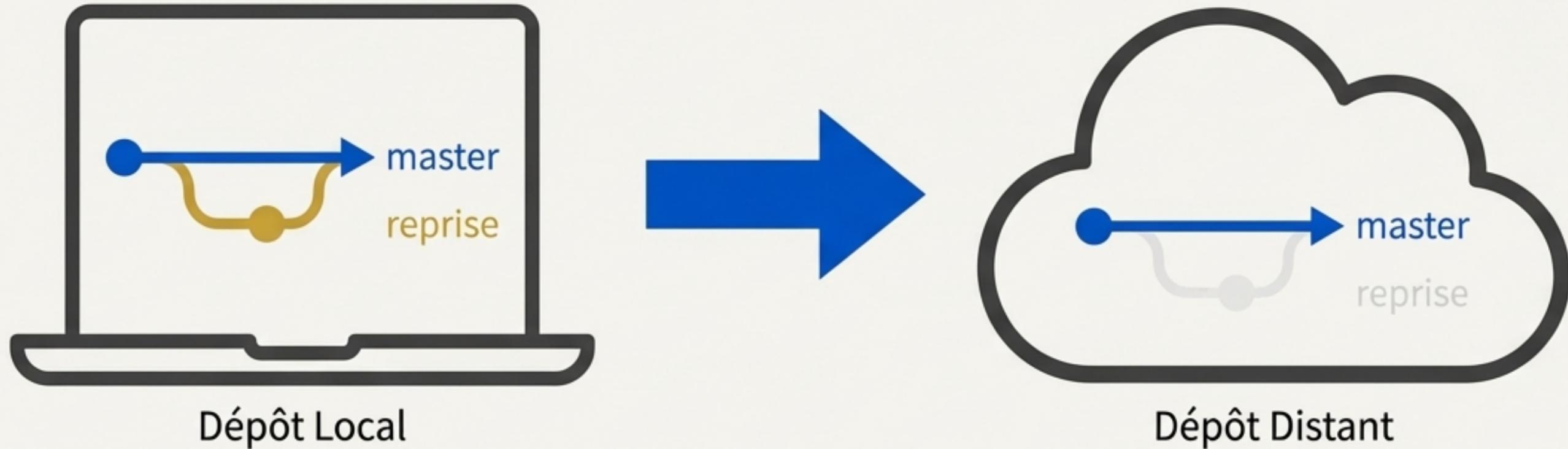
3. Vérifier l'historique (recommandé) :

```
git log --oneline --graph --all
```



**Votre branche `master` locale contient maintenant la version la plus à jour du projet.**

# Étape 5 : Mettre à Jour GitHub après la Fusion



Notre `master` local est à jour, mais GitHub ne le sait pas encore. La dernière étape est de synchroniser ce changement.

```
Pas besoin de spécifier la branche, car le lien de suivi a été établi
git push origin master
```

**\*\*Le dépôt distant reflète maintenant la fusion. La branche `master` sur GitHub est la version officielle et finale de notre travail.\*\***



## Étape 6 : Maintenir un Dépôt Propre (Optionnel)

Une fois qu'une branche de fonctionnalité a été fusionnée, elle n'est souvent plus nécessaire. La supprimer permet de garder un projet lisible.

### 1. Supprimer la branche en local :

```
git branch -d reprise
```

### 2. Supprimer la branche sur le dépôt distant (GitHub) :

```
git push origin --delete reprise
```

# Bilan des Commandes Clés



Voici la séquence de commandes qui résume notre mission, en excluant la publication initiale de `master` qui n'est faite qu'une seule fois :

```
Se placer sur la branche de travail (si besoin)
git switch reprise

Publication de la branche de travail
git push -u origin reprise

Passage sur la branche principale pour intégrer
git switch master

Intégration du travail
git merge reprise

Synchronisation du résultat final
git push origin master
```