

## En essayant de pousser mon travail sur git, j'ai rencontré cette erreur :

```
PS C:\wamp64\www\GeestionImmobolier\gestionImobillier> git push origin feature/new_branch_mountaga
remote: Permission to Adama-gueye/gestionImobillier.git denied to LamineTHIOUBE.
fatal: unable to access 'https://github.com/Adama-gueye/gestionImobillier.git/': The requested URL returned error: 403
PS C:\wamp64\www\GeestionImmobolier\gestionImobillier>
```

Après plusieurs recherches, je suis parvenu à résoudre ce problème.

### Erreur :

```
PS C:\wamp64\www\GeestionImmobolier\gestionImobillier> git push origin
feature/new_branch_mountaga remote : Autorisation d'Adama-gueye/gestionImobillier.git refusée à
LamineTHIOUBE. fatal : impossible d'accéder à 'https://github.com/Adama-
gueye/gestionImobillier.git/' : L'URL demandée a renvoyé l'erreur : 403 PS
C:\wamp64\www\GeestionImmobolier\gestionImobillier> Sur compte github, c'est Mountaga007 au
lieu de Lamine THIOUBE
```

### Solution :

Si vous rencontrez des problèmes d'autorisation et que le push est refusé vers le référentiel GitHub, il est probable que les informations d'identification enregistrées sur votre système soient associées au mauvais compte GitHub.

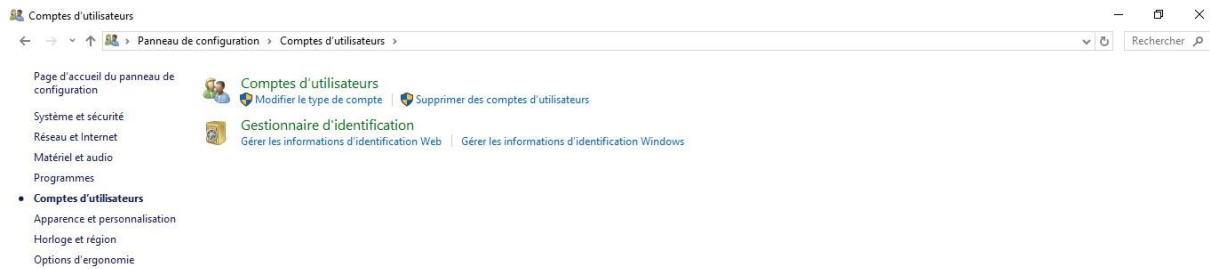
Pour résoudre ce problème :

1. **Mettre à jour les informations d'identification Git** : effacez les informations d'identification Git existantes pour forcer VS Code à demander de nouvelles informations d'identification.
  - Sous Windows, vous pouvez accéder au Credential Manager et supprimer toutes les informations d'identification liées à GitHub.

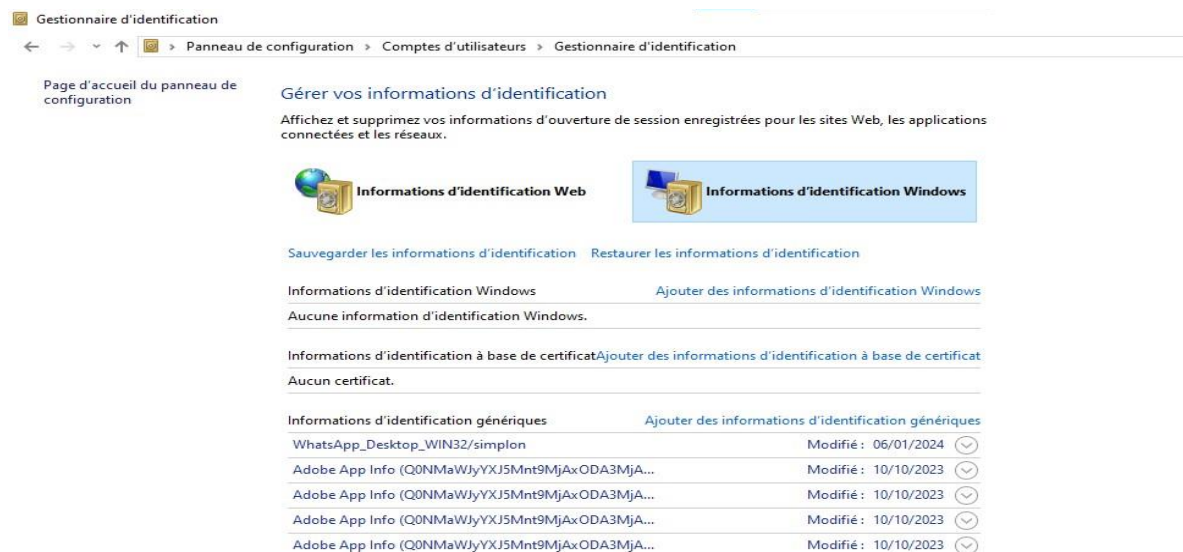
### Détail :

Pour effacer les informations d'identification Git sous Windows. Suivez ces étapes :

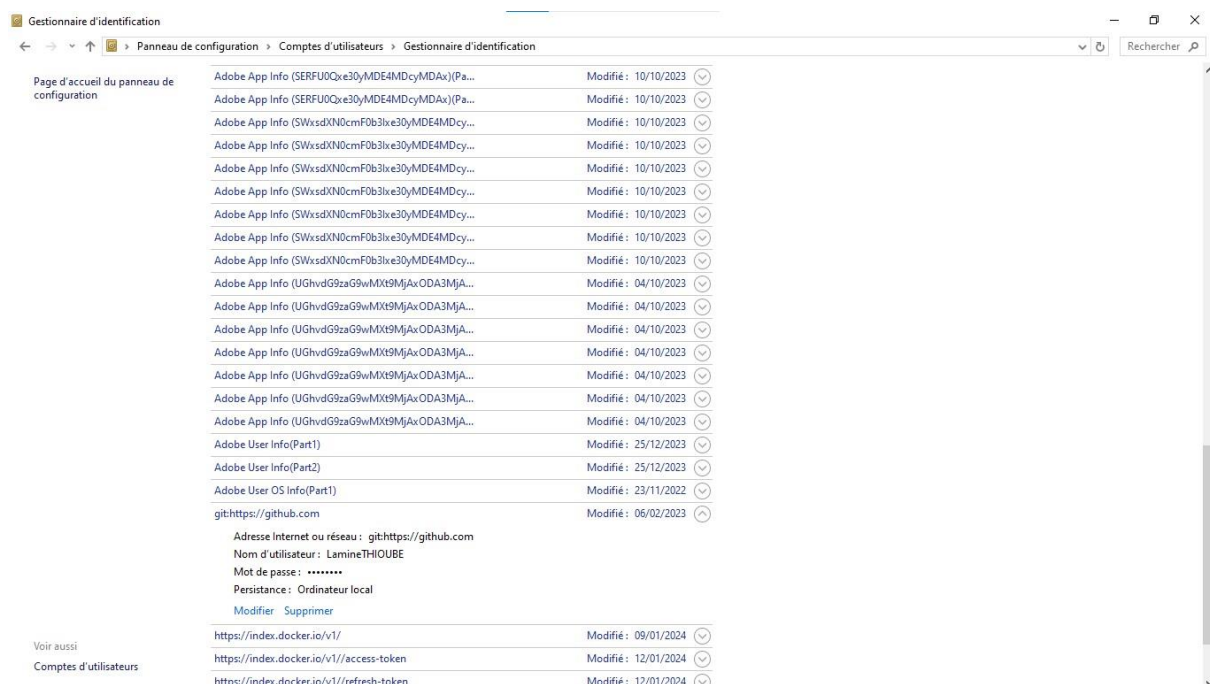
1. **Effacer les informations d'identification Git** : sous Windows, vous pouvez effacer les informations d'identification Git à l'aide de Credential Manager. Voici comment procéder :
  - Ouvrez le Panneau de configuration sur votre ordinateur Windows.



- Accédez à « Gestionnaire d'identification », puis sélectionnez « Information d'identification Windows ».



- Recherchez les informations d'identification liées à GitHub ou au référentiel avec lequel vous travaillez.



- Supprimez ces informations d'identification.

En faisant cela, vous forcez Git à demander des informations d'identification la prochaine fois que vous effectuerez une opération Git.

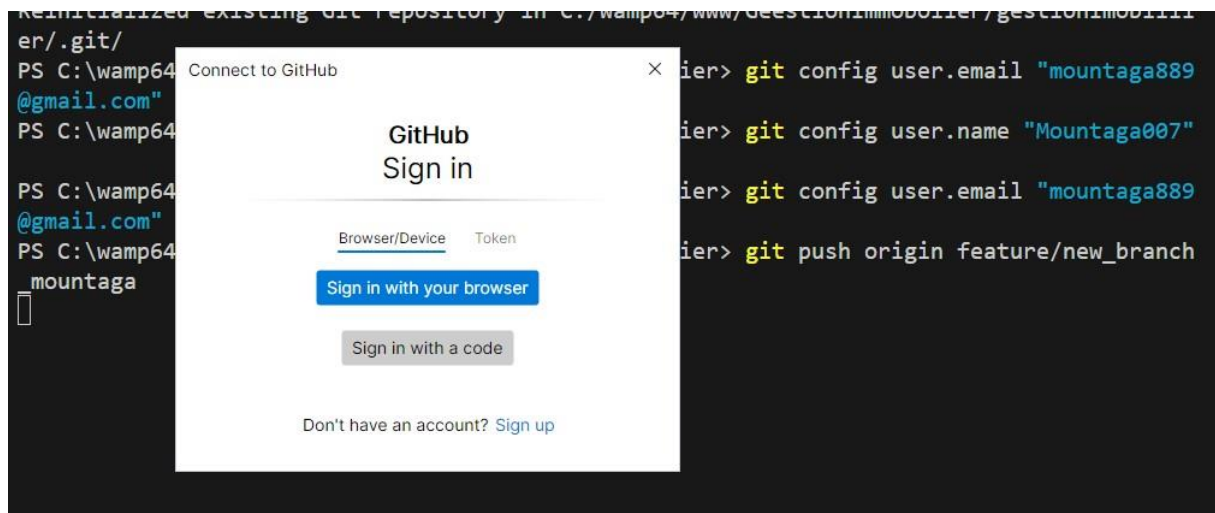
- **Pousser à nouveau les modifications** : essayez à nouveau de pousser vos modifications :

`git push origin nom_de_votre_branche` par exemple

`git push origin feature/new_branch_mountaga`

**Vous devriez être invité à saisir vos informations d'identification GitHub. Assurez-vous de saisir les informations d'identification associées au compte GitHub « Mountaga007 ».**

Ces étapes visent à garantir que vous utilisez les informations d'identification correctes et que Git vous demande le nom d'utilisateur et le mot de passe corrects si nécessaire.



- **Redémarrez Visual Studio Code** : après avoir effacé les informations d'identification, fermez et rouvrez Visual Studio Code.

Après avoir parvenu à pousser mon travail sur le repository distant « Adama-gueye », j'ai rencontré une autre erreur. En effet sur le repository, c'est le compte github de Lamine THIOUBE qui apparait au lieu de mon compte « Mountaga007 ».

Pour résoudre ce problème suivez les étapes suivantes :

**Erreur** : le commit est fait avec Lamine THIOUBE au lieu de Mountaga007. Voici ce que je vois sur le repository distant de Adama gueye après le **commit "LamineTHIOUBE Docker et test unitaire(créer un bien)"**.

Si le commit affiche un mauvais nom d'utilisateur, cela signifie que la configuration Git locale pour le nom et l'adresse e-mail de l'utilisateur n'est peut-être pas mise à jour. Voici comment vous pouvez le mettre à jour :

1. **Mettre à jour la configuration Git locale** : ouvrez le terminal intégré dans Visual Studio Code ou tout autre terminal que vous utilisez. Exécutez les commandes suivantes :

`git config user.name "Mountaga007"`

```
git config user.email "mountaga@example.com"
```

Remplacez "**Mountaga007**" et "**mountaga@example.com**" par le nom d'utilisateur et l'adresse e-mail GitHub corrects associés au compte GitHub.

2. **Modifier le dernier commit** : si vous venez d'effectuer un commit avec un mauvais nom d'utilisateur, vous pouvez utiliser l' **--amend** option pour modifier le dernier commit. Exécutez la commande suivante :

```
git commit --amend --reset-author -m "Votre nouveau message de commit"
```

3. **Forcer la validation mise à jour** : après avoir modifié la validation, vous devrez forcer la transmission des modifications au référentiel distant. Exécutez la commande suivante :

```
git push origin nom_de_votre_branche --force par exemple
```

```
git push origin feature/new_branch mountaga --force
```

Soyez prudent lorsque vous forcez, surtout si d'autres personnes travaillent sur la même branche. La poussée forcée réécrit l'historique, et il est généralement recommandé de l'éviter sur les branches partagées.

Avec ces étapes, vous devriez avoir le nom d'utilisateur correct associé au dernier commit, et cela devrait se refléter sur le référentiel distant.

N'oubliez pas de remplacer les valeurs d'espace réservé par votre nom d'utilisateur et votre adresse e-mail GitHub réels.

**Résultat final :**

