
Lab GitHub : Gestion des Branches et Synchronisation avec un Dépôt Distant

Objectifs

- Créer un dépôt Git local
 - Ajouter des fichiers et valider les modifications
 - Gérer les branches (`version1` et `main`)
 - Fusionner les branches
 - Connecter un dépôt local avec un dépôt GitHub distant
 - Modifier l'URL du dépôt distant
 - Utiliser un token GitHub pour pousser les modifications
-

1. Initialisation du Dépôt Git Local

```
mkdir projet-isil3
cd projet-isil3
git init
```

```
git config --global user.email "you@example.com"
git config --global user.name "Your Name"
```

2. Ajout de Fichiers et Validation des Modifications

```
echo "Bienvenue dans mon projet ISI L3" > fichier.txt
echo "Ceci est un fichier de configuration" > config.cfg
git add .
git commit -m "Initialisation du projet avec fichier.txt et config.cfg"
```

#Vérifier le résultat

```
git log
git status
```

3. Création et Gestion des Branches

3.1 Créer et Basculer sur une Nouvelle Branche `version1`

```
git checkout -b version1
```

```
#Vérifier  
git branch
```

3.2 Modifier un Fichier dans la Branche `version1`

```
echo "Modification spécifique à version1" >> fichier.txt  
git add fichier.txt  
git commit -m "Ajout de modifications dans version1"
```

3.3 Fusionner `version1` avec `main`

```
git checkout main  
git merge version1  
git commit -m "Fusion de version1 avec main"
```

4. Connexion avec un Dépôt Distant sur GitHub

4.1 Créer un Dépôt sur GitHub

1. Aller sur **GitHub**
2. Créer un **nouveau dépôt** (ex: `mon-projet-isil3`)
3. **Ne pas** initialiser avec un `README.md`

4.2 Connecter le Dépôt Local avec GitHub

```
git branch -M master  
git remote add origin https://github.com/UTILISATEUR/mon-projet-isil3.git
```

4.3 Modifier l'URL du Dépôt Distant

```
git remote set-url origin https://github.com/TON-NOM-UTILISATEUR/mon-  
projet-isil3.git
```

Vérifier que l'URL est bien mise à jour :

```
git remote -v
```

5. Authentification avec un Token GitHub

5.1 Générer un Token GitHub

1. Aller dans **GitHub** → **Settings** → **Developer settings** → **Personal access tokens**
2. **Générer un nouveau token (classic)**
3. Sélectionner les permissions :
 - o `repo` → Accès en lecture/écriture
 - o `workflow` (si besoin)
4. **Copier le token** (il ne sera plus affiché après)

5.2 Pousser le Code sur GitHub avec le Token

```
git push origin master
```

Lors de l'authentification, entrer **le token** à la place du mot de passe.

6. Vérification

1. Aller sur **GitHub** et vérifier que les fichiers sont bien présents.
 2. Confirmer que la branche `version1` a bien été fusionnée dans `main`.
-

7. Bonus : Ajouter un Fichier `README.md`

```
echo "# Mon Projet Git" > README.md
git add README.md
git commit -m "Ajout du README"
git push origin main
```

8. Annexe :

1-) Mettre à jour la branche `version1` sur GitHub

2-) Inviter un nouveau membre de l'équipe dans votre repository.

En tant qu'invité, Faire une pull request dans GitHub pour mettre à jour la branche `version1` à partir de la branche `main` .

En tant que propriétaire du repository valider la pull request et faire un merge..