

Guide Pratique de Git

Riyad Derguini

January 17, 2025

1 Introduction

Git est un système de contrôle de version distribué qui permet de suivre les modifications apportées à un ensemble de fichiers. Il est particulièrement utile pour la collaboration et la gestion de versions dans des projets logiciels.

2 Concepts de Base

2.1 Répertoire de Travail (Worktree)

Le répertoire de travail est l'endroit où se trouvent tous les fichiers que vous souhaitez versionner. Il peut contenir des sous-répertoires.

2.2 Dépôt Local (Repository)

Le dépôt local est la base de données qui contient l'historique complet des versions (commits). Il est stocké dans le répertoire caché `.git/` à la racine du répertoire de travail.

2.3 Staging Area (Index)

La staging area est une zone intermédiaire où vous préparez les fichiers pour le prochain commit. Vous pouvez ajouter, supprimer ou renommer des fichiers dans cette zone.

3 Commandes de Base

3.1 Créer un Dépôt

```
git init
```

3.2 Ajouter des Fichiers à la Staging Area

```
git add <filename>
git rm <filename>
git mv <filename> <new_filename>
```

3.3 Enregistrer un Commit

```
git commit -m "<message>"
```

4 Visualisation et Informations

4.1 Statut du Répertoire de Travail

```
git status
```

4.2 Différences entre les Zones

```
git diff
git diff --staged
git diff HEAD
```

4.3 Historique des Commits

```
git log
git log --graph --oneline --all --decorate --color
```

5 Annuler des Changements

5.1 Annuler des Modifications dans le Répertoire de Travail

```
git restore <filename>
```

5.2 Annuler des Modifications dans la Staging Area

```
git restore --staged <filename>
```

5.3 Annuler des Modifications dans les Deux Zones

```
git restore --staged --worktree <filename>
```

6 Branches et Tags

6.1 Créer une Branche

```
git branch <branch_name>
```

6.2 Changer de Branche

```
git switch <branch_name>
```

6.3 Créer un Tag

```
git tag <tag_name>
```

7 Travail avec des Dépôts Distants

7.1 Cloner un Dépôt Distant

```
git clone <remote>
```

7.2 Récupérer des Modifications d'un Dépôt Distant

```
git fetch <remote> <branch>
```

7.3 Fusionner des Modifications

```
git merge <branch>
```

7.4 Pousser des Modifications vers un Dépôt Distant

```
git push <remote> <branch>
```

8 Concepts Avancés

8.1 Fusionner des Branches

```
git merge <other_branch>
```

8.2 Choisir des Commits Spécifiques

```
git cherry-pick <commit>
```

8.3 Annuler un Commit

```
git revert <commit>
```

8.4 Ré-écrire l'Histoire

```
git rebase <branch>
```

9 Conclusion

Git est un outil puissant pour la gestion de versions et la collaboration. En comprenant les concepts de base et en maîtrisant les commandes essentielles, vous pouvez efficacement gérer vos projets et collaborer avec d'autres développeurs.