

Référence des Commandes Git

Gabriel Nomo

October 12, 2024

Commandes Git essentielles pour la collaboration

| Catégorie | Commande | Description |
|--------------------------------|--|---|
| Configuration | <code>git config --global user.name "Ton Nom"</code> | Configure ton nom d'utilisateur globalement. |
| | <code>git config --global user.email "ton@mail.com"</code> | Configure ton adresse e-mail pour tous les dépôts. |
| | <code>git config --list</code> | Vérifie les configurations actuelles. |
| Initialisation & Dépôt | <code>git init</code> | Initialise un dépôt Git dans le répertoire actuel. |
| | <code>git clone <url></code> | Clone un dépôt distant en local. |
| Gestion des Fichiers | <code>git add .</code> | Ajoute tous les fichiers modifiés au prochain commit. |
| | <code>git add <fichier></code> | Ajoute un fichier spécifique au prochain commit. |
| | <code>git rm <fichier></code> | Supprime un fichier du suivi par Git. |
| Commits | <code>git commit -m "message"</code> | Crée un commit avec un message descriptif. |
| | <code>git commit --amend</code> | Modifie le dernier commit. |
| Visualisation de l'État | <code>git status</code> | Montre l'état des fichiers suivis et non suivis. |
| | <code>git diff</code> | Compare les modifications non ajoutées au suivi. |
| | <code>git log</code> | Affiche l'historique des commits. |
| Branches | <code>git branch</code> | Liste toutes les branches locales. |
| | <code>git branch -a</code> | Liste toutes les branches locales et distantes. |
| | <code>git checkout -b <nom-branche></code> | Crée et bascule vers une nouvelle branche. |
| Fusion & Rebasage | <code>git merge <branche></code> | Fusionne une branche dans la branche active. |
| | <code>git rebase <branche></code> | Réapplique les commits par-dessus une autre. |
| Récupération & Synchronisation | <code>git fetch</code> | Récupère les modifications sans fusionner. |
| | <code>git pull origin main</code> | Récupère et fusionne les changements depuis main. |

Conseils pour la Collaboration

- **Travaille toujours sur une branche** : Utilise des branches pour isoler tes changements.
- **Utilise `git pull --rebase`** : Garde un historique propre.
- **Commits fréquents et clairs** : Chaque commit doit avoir un message descriptif.
- **Communique avec ton équipe** : Utilise les Pull Requests pour les validations.

| Consigne | Commande Git |
|---|--|
| Créer une nouvelle branche pour chaque fonctionnalité ou correction | <code>git checkout -b <nom-branche></code> |
| Ajouter les fichiers modifiés au suivi | <code>git add <fichier></code> ou <code>git add .</code> |
| Commiter les changements avec un message descriptif | <code>git commit -m "description du changement"</code> |
| Pousser les changements vers le dépôt distant | <code>git push origin <nom-branche></code> |
| Récupérer les changements du dépôt distant avant de pousser | <code>git pull origin <nom-branche></code> |
| Résoudre les conflits de fusion | <code>git merge <nom-branche></code> et gérer les conflits dans les fichiers concernés |
| Effectuer une revue de code via une Pull Request | <code>git request-pull <branche></code> |
| Synchroniser votre branche avec la branche principale | <code>git rebase <branche-principale></code> |