

Communauté et bonnes pratiques sur Gitlab

Une communauté GitLab est un lieu d'échanges où les développeurs partagent des expériences et des conseils. Les bonnes pratiques Git incluent le respect des conventions de nommage, une structure de branches claire et des messages de commit informatifs, contribuant ainsi à un environnement de développement collaboratif et organisé. De nombreuses recommandations existent et peuvent dépendre du langage informatique utilisé pour les codes, même s'il y a des bonnes pratiques communes, par exemple sur ce que doit être un bon commit.

De manière générale, un bon commit est essentiel pour maintenir un historique propre et compréhensible dans un projet Git. Voici quelques principes clés pour créer des commits de qualité :

- **Atomicité** : Un bon commit est atomique, ce qui signifie qu'il ne concerne qu'une seule modification cohérente. Évitez de mélanger plusieurs changements distincts dans un seul commit. Par exemple, si vous corrigez un bug et ajoutez une nouvelle fonctionnalité, créez deux commits séparés.
- **Clarté du message** : Le message de commit doit être clair et informatif. Il devrait expliquer brièvement ce que fait le commit. Évitez les messages vagues comme "Fix" ou "Update". Précisez plutôt ce qui a été modifié et pourquoi.
- **Résumé concis** : Le premier paragraphe du message de commit doit être un résumé concis (idéalement moins de 50 caractères) qui décrit l'essence du changement. Cela permet aux autres développeurs de comprendre rapidement le but du commit.
- **Description détaillée** : Après le résumé, ajoutez une description plus détaillée du changement. Expliquez le contexte, les problèmes résolus et les décisions prises. Utilisez un langage clair et précis.
- **Orthographe et grammaire** : Prenez le temps de relire votre message de commit. Des erreurs d'orthographe ou de grammaire peuvent rendre la compréhension difficile.
- **Évitez les modifications inutiles** : Ne committez pas de fichiers temporaires, de fichiers générés ou de modifications non pertinentes. Assurez-vous que votre commit ne contient que des changements significatifs. Pour limiter le risque de versionner des fichiers non souhaités, il est possible d'en déclarer certains selon des modèles de nommage. Des ressources en ligne permettent de lister par environnement de développement ou par langage une liste de fichiers à ignorer, par exemple : <https://gitignore.io>

En suivant ces principes, vous contribuerez à un historique Git propre et facilement compréhensible pour vous-même et vos collaborateurs.

Pour acquérir les compétences pratiques pour utiliser GitLab avec Git de manière efficace, consultez les supports de formation des Centre INRAE Provence-Alpes-Côte d'Azur.

<https://git-gitlab-paca.pages.mia.inra.fr/support-de-formation/>

Pour citer cette fiche :

INRAE DipSO (2024). Communautés et bonnes pratiques sur Gitlab. Formation OSCAR Ouvrir la science, connaissance à acquérir : module Gestion et partage des algorithmes, codes et logiciels. 2 p.

oscar.inrae.fr