

# CONVENTION DE COMMIT GIT

## Exemples de format de message conventionnel

Type[fonctionnalité] : commitSubject -> Description

ex : feat[Print Dashboard]: besoin d'afficher la liste prochains départ de train -> récupération et affichage de la liste de train prévu pour la journée.

## Format de message de validation conventionnel

Nous utiliserons l'engagement conventionnel pour faire respecter la discipline.

Le message de validation doit être structuré comme suit :

1. Le header est obligatoire et la portée du header est facultative.
2. Une ligne du message de validation ne peut pas dépasser 120 caractères ! Cela permet au message d'être plus facile à lire sur GitHub ainsi que dans divers outils git.

ex : feat[Print Dashboard]

## Type de commit conventionnel

Le type Doit être l'un des suivants :

**build** : modifications qui affectent le système de génération ou les dépendances externes (exemples de portées : webpack, karma, npm)

**ci** : Modifications apportées à nos fichiers et scripts de configuration CI (exemples de scopes : azuredevops.yaml, Travis, Circle, BrowserStack, SauceLabs)

**docs** : modifications de la documentation uniquement

**feat** : Une nouvelle fonctionnalité

**fix** : une correction de bug

**perf** : un changement de code qui améliore les performances

**refactor** : modification du code qui ne corrige pas un bug et n'ajoute pas de fonctionnalité

**style** : modifications qui n'affectent pas la signification du code (espace, mise en forme, points-virgules manquants, etc.)

**test** : ajout de tests manquants ou correction de tests existants

## Objet de validation conventionnel

Le **subject** contient une description succincte du changement et voici quelques conseils pour le texte du **subject**:

- Utilisez l'impératif, au présent : « changement » et non « changé » ni « changements »
- Ne mettez pas de majuscule à la première lettre

pas de point (.) à la fin

## Organe de validation conventionnel

- Tout comme dans le **subject**, utilisez l'impératif, le présent : « changement » et non « changé » ni « changements ».
- Le **body** doit inclure la motivation du changement et la comparer avec le comportement précédent.
- Le **footer** doit contenir toutes les informations sur les modifications de rupture et est également l'endroit pour référencer les problèmes GitHub que cette validation ferme.
- **Breaking Changes** doit commencer par le mot **BREAKING CHANGE** : avec un espace ou deux nouvelles lignes. Le reste du message de validation est ensuite utilisé à cette fin.

source : <https://medium.com/rupesh-tiwari/conventional-git-commit-messages-and-linting-76e1fbb9e14a>