

# git stash

Git stash est une commande de Git qui permet de "**mettre de côté**" des modifications. Cette fonctionnalité est très pratique, notamment lorsque le développeur a besoin de changer de branche rapidement. Cette fonctionnalité lui permet de ne pas se presser à effectuer un commit bâclé pour pouvoir changer de branche.

Git stash va récupérer les **modifications en cours** et les enregistrer dans un **conteneur** qu'on appellera un **stash**.

Pour obtenir des informations plus complètes sur git stash vous pouvez vous référer au livre git : maitrisez la gestion de vos versions.

## Sauvegarder les modifications dans un stash

Pour cela, il faut utiliser la commande suivante :

**git stash**

Il est également possible d'utiliser la commande complète :

**git stash save**

Pour intégrer un message personnalisé au stash il faut utiliser la syntaxe suivante (**WIP** est souvent utilisé pour dire **Work In Progress**) :

**git stash save "WIP add charts to mobile app"**

Par défaut, les fichiers non suivis (inconnus de Git) ne seront pas intégrés dans le stash, pour le faire il faut utiliser la commande suivante :

**git stash save --include-untracked**

## Lister les stashes enregistrés

Pour lister tous les stashes enregistrés il faut utiliser la commande suivante :

**git stash list**

Les stashes seront alors listés de la façon suivante :

**stash@{0}: WIP add charts to mobile app**

**stash@{1}: WIP QR code integration**

Les noms de type **stash@{0}** sont des références qui pourront être utilisées.

## Récupérer les données du stash

Pour réintégrer dans le répertoire de travail et l'index les données contenues dans le dernier stash il faut utiliser la commande suivante :

## **git stash pop**

Il est également possible de spécifier une référence de stash pour réintégrer un stash précis et non le dernier :

### **git stash pop stash@{0}**

Où **stash@{0}** est la référence du stash qu'on souhaite intégrer.

Par défaut, la commande **git stash pop** va supprimer le stash une fois que celui-ci aura été réintégré. Pour conserver malgré tout le stash, il faut utiliser la commande **git stash apply** avec les mêmes arguments que le commande **git stash pop**.

## **Afficher les données d'un stash**

Lorsqu'on enregistre pas mal de stash sans message (ce n'est pas bien ! Bouh !) il arrive fréquemment qu'on se perde dans la liste des stashes. Pour afficher les données d'un stash (et enfin s'y retrouver) il faut utiliser la commande suivante :

### **git stash show stash@{0}**

Où **stash@{0}** est la référence du stash qu'on souhaite inspecter.