

# GIT FLOW

## Premiers pas

Git flow doit être initialisé afin de personnaliser la configuration de votre projet.

## Initialisation

Commencez à utiliser git-flow en l'initialisant dans un dépôt git existant :

### **git flow init**

Vous devrez répondre à quelques questions concernant les conventions de nommage pour vos branches.

Il est recommandé d'utiliser les valeurs par défaut.

## Fonctionnalités/Features

- Développe des nouvelles fonctionnalités pour la prochaine version
- Existe en général uniquement dans les dépôts des développeurs

## Commencer une feature

Le développement d'une fonctionnalité commence à partir de la branche '**develop**'

Commencer le développement d'une nouvelle fonctionnalité avec :

### **git flow feature start MYFEATURE**

Cette commande crée une nouvelle branche de fonctionnalité basée sur '**develop**' et passe sur cette branche

## Terminer une fonctionnalité

Cette action effectue les opérations suivantes:

- Fusionne MYFEATURE dans 'develop'
- Supprime la branche de fonctionnalité
- Passe sur la branche 'develop'

### **git flow feature finish MYFEATURE**

## Publier une fonctionnalité

Publiez une fonctionnalité sur le serveur distant pour qu'elle puisse être utilisée par d'autres utilisateurs.

### **git flow feature publish MYFEATURE**

## Récupérer une fonctionnalité publiée

Récupérer une fonctionnalité publiée par un autre utilisateur

### **git flow feature pull origin MYFEATURE**

Vous pouvez suivre une fonctionnalité sur le serveur distant en utilisant **git flow feature track MYFEATURE**

## Livraison/Release

- Prépare la sortie d'une nouvelle version de production
- Permet les corrections de bugs mineurs et la préparation des métadonnées de la release

## Commencer une livraison

Pour commencer une livraison, utilisez la commande **git-flow release**

Créer une branche de livraison basée sur la branche de développement.

### **git flow release start RELEASE [BASE]**

Vous pouvez si besoin ajouter le paramètre [BASE], le hash d'un commit à partir duquel commencera la livraison. Ce commit doit faire partie de la branche de développement.

Il est préférable de publier la branche de livraison après l'avoir créée pour permettre aux autres développeurs de commiter dessus. De la même manière que pour les fonctionnalités, utilisez cette commande:

### **git flow release publish RELEASE**

Vous pouvez suivre une livraison sur le serveur distant en utilisant

### **git flow release track RELEASE**

## Terminer une livraison

Plusieurs actions sont réalisées :

- Fusionne la branche de livraison dans la branche 'master'
- Etiquette la livraison par son nom
- Fusionne la livraison dans la branche 'develop'
- Supprime la branche de livraison

### **git flow release finish RELEASE**

N'oubliez pas de pousser vos étiquettes avec `git push --tags`

## Correctifs/Hotfixes

- Les correctifs sont utiles quand il est nécessaire de corriger immédiatement l'état incorrect de la version en production
- Ils peuvent se baser sur l'étiquette de la branche '**master**' indiquant la version en production.

### Commencer un hotfix

Comme pour les autres commandes git-flow, un hotfix est commencé par

**git flow hotfix start VERSION [BASENAME]**

Ici, le paramètre **VERSION** indique le nom de la future release corrigée. Vous pouvez si besoin spécifier à quelle release s'appliquera le **hotfix**.

### Terminer un hotfix

En terminant un **hotfix**, il est fusionné dans les branches '**develop**' et '**master**'. De plus la fusion vers '**master**' est étiquetée par la version du **hotfix**.

**git flow hotfix finish VERSION**

### Commandes

