

TP Git Flow

Principes de base

Le principe de base de fonctionnement est d'avoir 2 branches principales

- **master**, est la branche qui sera reliée à la mise en production de votre application.
- **develop**, est une branche sur laquelle tous les développeurs vont travailler.

Mais si tout le monde commence à travailler sur la même branche, l'historique peut rapidement devenir fouilli et les modifications risquent de se court-circuiter. Ainsi on a second niveau de branche :

- Les branches **feature/XXXX** permettent de commencer à travailler sur une nouvelle fonctionnalité. On crée une nouvelle branche à partir de la branche develop, on fait nos commits dessus et quand on finit on fusionne sur la branche develop.
- Les branches **release/XXXX** permettent de faire la liaison entre la branche develop et la branche. En général cette branche contiendra des commits pour préparer une nouvelle release (on va changer le numéro de version dans le fichier readme par exemple).
- Les branches **hotfix/XXXX** permettent de publier rapidement une correction depuis la branche master. Une fois la correction validée nos modifications seront fusionnées sur la branche master et develop.

Créer un nouveau projet sur GitHub

Par la suite, veuillez créer un nouveau répertoire avec les commandes suivantes :

git clone ...

Aller dans votre nouveau répertoire :

cd repogitflow

Créer un nouveau fichier :

touch README.md

Ajouter, commiter puis pusher sur le serveur distant :

```
git add README.md
```

```
git commit -m "add README"
```

```
git push -u origin master
```

Lister vos branches :

```
git branch
```

Installer Git flow si besoin (Linux) :

```
sudo apt get install git-flow
```

Initialiser et paramétrer vos branches avec la commande suivante :

```
git flow init
```

Branch name for production releases: [master]

Branch name for "next release" development: [develop]

How to name your supporting branch prefixes?

Feature branches? [feature/]

Bugfix branches? [bugfix/]

Release branches? [release/]

Hotfix branches? [hotfix/]

Support branches? [support/]

Version tag prefix? []

Hooks and filters directory? [C:/Users/Dell/Desktop/RepoGitFlow/.git/hooks]

Appuyer sur “Enter” à chaque nouvelle entrée, il s’agit du paramétrage de notre Workflow.

Comme vous pouvez le constater, plusieurs branches et sous branches vont être créés.

Lister vos branches :

```
git branch
```

```
* develop
```

```
master
```

Nous constatons la création de la branche **develop**

Commençons le développement d'une nouvelle fonctionnalité avec :

git flow feature start homepage

Switched to a new branch 'feature/homepage'

Summary of actions:

- A new branch 'feature/homepage' was created, based on 'develop'

- You are now on branch 'feature/homepage'

Now, start committing on your feature. When done, use:

git flow feature finish homepage

Lister vos branches :

git branch

develop

* feature/homepage

master

Créer un nouveau fichier index.html :

vi index.html

Ajouter, commiter puis pusher sur le serveur distant :

git add index.html

git commit -m "add index.html"

git push origin feature/homepage

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

git flow feature publish homepage

Récupérer une fonctionnalité publiée par un autre utilisateur

git pull origin feature/homepage

Terminer la fonctionnalité

git flow feature finish homepage

Switched to branch 'develop'
Updating 328b454..94534fb
Fast-forward
index.html | 0
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 index.html
To gitlab.com:DamienMonchaty/repogitflow.git
- [deleted] feature/homepage
Deleted branch feature/homepage (was 94534fb).

Summary of actions:

- The feature branch 'feature/homepage' was merged into 'develop'
- Feature branch 'feature/homepage' has been locally deleted; it has been remotely deleted from 'origin'
- You are now on branch 'develop'

Cette action effectue les opérations suivantes:

- Fusionne Feature/homepage dans 'develop'
- Supprime la branche de fonctionnalité
- Passe sur la branche 'develop'

Affichons notre historique :

git log --oneline --graph --color --all --decorate

* 94534fb (HEAD -> develop) add index.html
* 328b454 (origin/master, master) init commit

Commencer une livraison

git flow release start v0.0.1

Switched to a new branch 'release/v0.0.1'

Summary of actions:

- A new branch 'release/v0.0.1' was created, based on 'develop'
- You are now on branch 'release/v0.0.1'

Follow-up actions:

- Bump the version number now!
- Start committing last-minute fixes in preparing your release
- When done, run:

git flow release finish 'v0.0.1'

Modifiez le fichier README en y insérant le texte

Version :0.0.1

Vi README.md

Ajouter, commiter :

```
git add README.md
```

```
git commit -m "add bump version"
```

Terminer la livraison

```
git flow release finish v0.0.1
```

Une fenêtre s'ouvrira et vous demandera de rentrer un message correspondant à un tag, insérer

v0.0.1

Taper esc , puis :wq

```
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
Merge made by the 'recursive' strategy.
 README.md | 1 +
 index.html | 0
 2 files changed, 1 insertion(+)
 create mode 100644 index.html
Already on 'master'
Your branch is ahead of 'origin/master' by 3 commits.
(use "git push" to publish your local commits)
Switched to branch 'develop'
Merge made by the 'recursive' strategy.
 README.md | 1 +
 1 file changed, 1 insertion(+)
Deleted branch release/v0.0.1 (was 625e7d9).
```

Summary of actions:

- Release branch 'release/v0.0.1' has been merged into 'master'
- The release was tagged 'v0.0.1'
- Release tag 'v0.0.1' has been back-merged into 'develop'
- Release branch 'release/v0.0.1' has been locally deleted
- You are now on branch 'develop'

Envoyons sur le repository distant le tag que nous venons de créer

```
git push --tags
```

```
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 4 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (7/7), 724 bytes | 241.00 KiB/s, done.
Total 7 (delta 2), reused 0 (delta 0)
To gitlab.com:DamienMonchaty/repogitflow.git
 * [new tag]      v0.0.1 -> v0.0.1
```

Mettez à jour vos remotes

git push origin master

git push origin develop

Commencer un hotfix

Imaginons nous nous étions trompés dans le titre de notre page index, nous aurons mis en place un branche de correctif comme ceci :

git flow hotfix start title

Branches 'master' and 'origin/master' have diverged.
And local branch 'master' is ahead of 'origin/master'.
Switched to a new branch 'hotfix/title'

Summary of actions:

- A new branch 'hotfix/title' was created, based on 'master'
- You are now on branch 'hotfix/title'

Follow-up actions:

- Start committing your hot fixes
- Bump the version number now!
- When done, run:

git flow hotfix finish 'title'

Modifiez le fichier index.html en y insérant le titre

```
<title>Premier Git Flow</title>
```

vi index.html

Ajouter, commiter :

git add index.html

git commit -m "changement du titre"

Terminer un hotfix

git flow hotfix finish title

Une fenêtre s'ouvrira et vous demandera de rentrer un message correspondant à cet Hotfix, insérer

Hotfix title

Taper esc , puis :wq

Lister vos branches :

git branch

* develop

master

Taper **gitk** pour voir votre historique !!! Pour pouvoir continuer, veuillez fermer la fenêtre

Affichons notre historique :

git log --oneline --graph --color --all --decorate

```
hell@DamPc MINGW64 ~/Desktop/RepoGitFlow (develop)
git log --oneline --graph --color --all --decorate
* fb79b2f (HEAD -> develop, origin/develop) Merge tag 'title' into develop
/
* 5d47949 (tag: title, master) Merge branch 'hotfix/title'
|
* 2fd3415 changement du titre
|
/
* 3236a5c Merge tag 'v0.0.1' into develop
|
/
* b20f838 (tag: v0.0.1) Merge branch 'release/v0.0.1'
|
* 625e7d9 bump version
|
/
* 94534fb add index.html
/
* 328b454 (origin/master) init commit
```

Pour afficher l'historique d'une seule branche, veuillez taper la commande suivante :

Git log --graph --oneline --first-parent develop

Commençons le développement d'une nouvelle fonctionnalité avec :

git flow feature start demo

Modifier trois fois le fichier index.html :

vi index.html

et faire trois commits correspondants :

git commit -a -m "a1"

git commit -a -m "a2"

git commit -a -m "a3"

Terminer la fonctionnalité

git flow feature finish demo

Switched to branch 'develop'

Merge made by the 'recursive' strategy.

index.html | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)

Deleted branch feature/demo (was b84dd83).

Summary of actions:

- The feature branch 'feature/demo' was merged into 'develop'
- Feature branch 'feature/demo' has been locally deleted
- You are now on branch 'develop'

Affichons notre historique :

git log --oneline --graph --color --all --decorate

```
* 773bd8f (HEAD -> develop) Merge branch 'feature/demo' into develop
|\
* b84dd83 a3
* 15bbdbd a2
* 4df29c3 a1
|/
* fb79b2f (origin/develop) Merge tag 'title' into develop
|\
* 5d47949 (tag: title, master) Merge branch 'hotfix/title'
| \
* 2fd3415 changement du titre
|/
* 3236a5c Merge tag 'v0.0.1' into develop
| \
| \
* b20f838 (tag: v0.0.1) Merge branch 'release/v0.0.1'
| \
* 625e7d9 bump version
|/
|/
* 94534fb add index.html
|/
* 328b454 (origin/master) init commit
```

Finir par :

git push origin develop

FIN

Exercices :

Selon le TP, veuillez dans le meme projet à la suite :

- **Créer une nouvelle release v0.0.2**
- **Créer un nouvelle hotfix, imaginons aue nous nous somme encore trompé dans le titre de la page index par exemple.**
- **Créer une nouvelle fonctionnalité appelé « contactpage ».**
- **Créer une nouvelle release v0.0.3.**