

git-collaboratif-depot-unique

Alice et Bob commencent à collaborer avec Git	1
paramétrages en vue de collaborer	1
Dépôt initial et invitations	1
Bob clone le projet d'Alice	1
remotes	1
Réalisation d'une fonctionnalité par Alice	2
Alice pousse son main sur son remote	2
Bob collabore avec Alice	2
Bob met à jour son dépôt local avec la dernière version du projet	2
Bob réalise à son tour une fonctionnalité décrite dans un item de backlog	2
Alice récupère le travail de Bob	2

Alice et Bob commencent à collaborer avec Git

paramétrages en vue de collaborer

Le but de Git est de collaborer. Faisons donc travailler ensemble Alice et Bob (et Gustave si besoin) dans un projet

Dépôt initial et invitations

Alice a créé un dépôt initial sur Gitlab qui s'appelle monprojet. Elle va inviter Bob sur son projet en tant que developer (droit de lecture et écriture) ou maintenir (tous les droits).

Bob clone le projet d'Alice

Bob va faire un clone du dépôt d'Alice sur Gitlab, pour obtenir une copie locale :

```
git clone https://gitlab.com/alice/monprojet.git
```

remotes

- Bob possède donc un remote origin correspondant au dépôt de référence d'Alice.
- Il peut le vérifier avec la commande `git remote -v`

Bob peut maintenant collaborer avec Alice sur ce dépôt commun.

Réalisation d'une fonctionnalité par Alice

- Alice prend une chose à réaliser (item de backlog) et implémente le code nécessaire
- Alice fait les tests et vérifie que ça marche
- `git commit -am \"Message de commit\"`

Alice pousse son main sur son remote

```
git push -u origin main
```

Bob collabore avec Alice

Bob fait d'abord comme Alice pour paramétrer son git et son dépôt local. Bob et Alice vont travailler tout d'abord sur la branche main. Cela peut sembler plus simple mais en fait on va passer rapidement à un travail dans des branches spécifiques à chaque tâche et utiliser des Merge Request (MR ou Pull Requests, PR sur Github) :

[Git avec les branches](#)

car c'est plus clair et finalement pas plus compliqué ! mais n'allons pas trop vite ...

Bob met à jour son dépôt local avec la dernière version du projet

```
git pull
```

Bob réalise à son tour une fonctionnalité décrite dans un item de backlog

```
git commit -m "fonctionnalité XXX réalisée"  
git push -u origin main
```

puis informe Alice qu'elle peut récupérer le code correspondant

Alice récupère le travail de Bob

```
git pull
```

Ce qui peut-être fait en deux temps également :

- Alice fetch le main de Bob pour se mettre à jour :

```
git fetch Bob main
```

- puis fusionne le code concerné (après l'avoir éventuellement contrôlé)

```
git merge Bob/main
```

2022 Gérard Rozsavolgyi roza [a] univ-orleans.fr