

DOCUMENT DE RÉFÉRENCE UTILISATION DE GIT ET GITHUB

Édouard LUMET <edouard.lumet@etu.enseeiht.fr>

12 juin 2018

À propos

Ce document de référence a pour but de décrire l'utilisation de Git en ligne de commande ainsi que l'organisation du dépôt accessible sur Github.

Git en ligne de commande

Ajout, suppression et déplacement de fichiers

Lors de la création de nouveaux fichiers, utilisez la commande git add :

```
$ git add <fichier> [...]
```

Lors de la suppression de fichiers, utilisez la commande git rm :

```
$ git rm <fichier> [...]
```

NB : cette commande supprime le fichier du système de fichiers

Lors du déplacement ou du renommage de fichiers, utilisez la commande git mv :

```
$ git mv <source> <destination>
```

NB : comme pour la commande précédente, l'action est répercutée sur le système de fichiers

Modification de fichiers

Lors du développement et de la modification de code, de fichiers, il faut ajouter lesdits fichiers à la base locale et effectuer le commit :

```
$ git add <fichier> [...]  
$ git commit -m "commentaire"
```

Une fois que les modifications sont ajoutées au commit (et éventuellement après plusieurs commits), il faut les envoyer sur le dépôt Github. Avant de se faire, il faut s'assurer d'être à jour :

```
$ git pull --rebase  
$ git push
```

NB : suite à la mise à jour de votre copie locale, vous pourrez avoir à gérer des conflits. Ceux-ci apparaîtront des les fichiers indiqués à l'aide marqueur ajoutés par Git (présence de chevrons avec des mots type HEAD, etc). En cas de conflit, n'hésitez pas à demander.

Bonnes pratiques

Comme indiqué précédemment, il est de mise d'effectuer un *pull -rebase* avant chaque envoi (*push*).

Il est aussi préférable d'effectuer des *pull -rebase* réguliers, en début de séance par exemple, pour démarrer sur une copie à jour.

Les commits devraient être atomiques. Pour rappel, leur longueur maximum préconisée est de 90 caractères. Pour ajouter des détails, exécuter la commande suivante à la place de celle indiquée plus haut :

```
$ git commit
```

Un éditeur s'ouvrira alors.

Organisation du dépôt Github

Deux branches ont été créées. La première est la principale, *master*. Celle-ci est protégée et seul l'administrateur peut y effectuer des commits ou un merge. La seconde est la branche de développement sur laquelle tout le monde travaille, à savoir *dev*.

Dans le dépôt, vous trouverez trois dossiers :

- *doc* : contient la documentation du projet dont la Javadoc, les diagrammes UML, etc,
- *iterations* : contient diverses informations à propos des sprints de la méthode Agile,
- *src* : contient les sources du projet (code Java).