

Directives sur le workflow

Benjelloun Otman
Birée Thomas
Boulouz Amine
El Goumiri Rida
Loginova Aleksandra

December 16, 2024

Ce PDF sera changé adapté au fur et à mesure.

Git

Pour éviter les aléas liés à l'infrastructure de l'ENSIMAG, nous utiliserons principalement un repo github personnel (privé) pour le développement: <https://github.com/otman-benjelloun/ProjetGL>. Pour un guide très complet sur les bases de Git, se référer au wiki des BugBusters (nécessite le VPN) : https://bugbusters.pages.ensimag.fr/wiki/linux_en_general/git/index.html

Git Flow

Nous utiliserons le workflow GitFlow, pour éviter au maximum les conflits Git, et avoir un développement cohérent et structuré: une documentation extensive est disponible ici: <https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow>. Il existe une extension pour gérer GitFlow (qui permet de ne pas avoir à créer et merge les branches manuellement), dont la documentation est présente ici: <https://github.com/nvie/gitflow>. Pour l'installer sur ubuntu :

```
sudo apt update \&\& sudo apt -y install git-flow
```

Après votre premier pull, et en ayant installé l'extention Git Flow, exécutez:

```
git flow init
```

Ensuite, garder toutes les valeurs par défaut (entrée jusqu'à la fin).

Messages de commit

Pour s'y retrouver dans l'historique des changements, essayer au plus de se tenir aux conventions de messages de commit détaillés ici: <https://www.freecodecamp.org/news/how-to-write-better-git-commit-messages/>. Principalement, pour un message de commit, le premier mot doit décrire le type de commit:

- **feat** a new feature is introduced with the changes
- **fix** a bug fix has occurred
- **chore** changes that do not relate to a fix or feature and don't modify src or test files (for example updating dependencies)
- **refactor** refactored code that neither fixes a bug nor adds a feature
- **docs** updates to documentation such as a the README or other markdown files
- **style** changes that do not affect the meaning of the code, likely related to code formatting such as white-space, missing semi-colons, and so on.
- **test** including new or correcting previous tests
- **perf** performance improvements

- **ci** continuous integration related
- **build** changes that affect the build system or external dependencies
- **revert** reverts a previous commit

Maven

TODO : se familiariser avec Maven.

Tests

TODO : trouver et se mettre d'accord sur la méthode d'intégration des tests.