

# TP4 – ARCHITECTURE WEB

## CONFLITS ET PLUS

### CONFLITS, FUSION CONFLICTUELLE

Afin de vous familiariser réellement avec les conflits de merge, on simulera un conflit à examiner et à résoudre ultérieurement.

- Créez un répertoire nommé git-merge-test, basculez vers ce répertoire, puis initialisez-le en tant que nouveau dépôt Git.
- Créez un fichier texte merge.txt avec du contenu.
- Ajoutez merge.txt au dépôt et committez-le.

Nous disposons à présent d'un nouveau dépôt avec une branche (main) et un fichier merge.txt avec du contenu. Nous allons maintenant créer une branche à utiliser en tant que merge conflictuel.

- Créer une branche nommée new\_branch et en faire un checkout
- Remplacer le contenu dans le fichier merge.txt
- Faire un commit du nouveau contenu

Avec cette nouvelle branche : new\_branch, nous avons créé un commit qui remplace le contenu du fichier merge.txt.

Notre exemple de dépôt se retrouve ainsi à un état où nous avons deux nouveaux commits. Un dans la branche (main) et un dans la branche new\_branch.

À présent, exécutons la commande git merge new\_branch pour voir ce qui se produit !

Un conflit apparaît. Et Git est tellement sympa qu'il nous en avertit !

### COMMENT IDENTIFIER LES CONFLITS DE MERGE ?

Pour en savoir plus, nous pouvons exécuter la commande git status.

La sortie de la commande git status indique que des chemins n'ont pas été mergés en raison d'un conflit. Le fichier merge.txt apparaît désormais à un état modifié. Examinons le fichier et découvrons ce qui a été modifié.

Nous pouvons voir certains ajouts étranges.

```
<<<<<< HEAD
=====
>>>>>> new_branch
```

Considérez ces nouvelles lignes comme des « séparateurs de conflit ».

La ligne ===== est le « centre » du conflit. Tout le contenu entre le centre et la ligne <<<<<< HEAD dans la branche principale actuelle vers laquelle pointe la réf HEAD. Autrement, tout le contenu entre le centre et >>>>>> new\_branch est présent dans notre branche de merge.

### COMMENT RESOUDRE DES CONFLITS DE MERGE A L'AIDE DE LA LIGNE DE COMMANDE ?

La manière la plus directe de résoudre un conflit de merge consiste à modifier le fichier conflictuel. Ouvrez le fichier merge.txt dans votre éditeur préféré. Dans notre exemple, supprimons simplement tous les séparateurs de conflit.

Une fois le fichier modifié, utilisez `git add merge.txt` pour stager le nouveau contenu mergé. Pour finaliser le merge, créez un commit.

Git voit que le conflit a été résolu et crée un commit de merge pour finaliser le merge.

## GITHUB PAGES

Vous trouverez une procédure complète d'utilisation des Github Pages à cette adresse :

<https://pages.github.com>

Créez un dépôt de la forme VOTREPSEUDO.github.io

Activez dans la partie Settings de ce dépôt, les GitHub pages, quelques explications sont disponibles ici : <https://docs.github.com/fr/pages/quickstart>

Déposez quelques pages html sur ce dépôt et vérifiez leur présence en ligne en utilisant comme url le nom de votre dépôt.

Vous pouvez créer des Github Pages de manière globale, à votre nom/pseudo, ou par individuellement par dépôt.