

Git

TP1 : merge et branches

Si vous avez un compte GitHub, vous pourrez l'utiliser. Sinon vous pouvez utiliser le Git de l'université : git.unilim.fr.

Rappel sur git:

Créer un nouveau dépôt :

git init

Cloner un dépôt :

git clone username@host:/path/to/repository

Indiquer le dépôt officiel :

git remote add official url/du/depot

Votre dépôt local est composé de 3 TREE :

- Working directory : votre répertoire de travail qui contient tous vos fichiers
- Index ou Stage area : espace de transit pour vos fichiers
- Head : pointe vers la dernière validation faite (le dernier envoi des données)

Créer une nouvelle branche et passer dessus pour l'utiliser :

git checkout -b nomBranche

Retourner sur la branche principale (master) :

git checkout master

Supprimer la branche :

git branch -d nomBranche

Mettre à jour le dépôt local :

git pull

Fusionner la branche active avec une autre :

git merge monAutreBranche

Git va tenter une fusion des changements, si conflits il faudra les résoudre et les marquer comme résolus : *git add nomFichier*

Visionner les changements après une fusion :

git diff nomBrancheSource nomBrancheCible

Savoir sur quelle branche on se trouve :

git branch

Vider l'index pour permettre un pull :

git reset --mixed

git reset --hard

Mise en application :

Utiliser votre compte GitHub et créer un nouveau dépôt.

En local, placez-vous dans un répertoire GitTP1, créer et cloner le dépôt.

Créer quelques fichiers xml ou text (fichier1, fichier2, fichier3) placez dans ces fichiers des listes de Nom Prénom en prenant soin à ce que vos 3 fichiers soient clairement différents.

Indiquez votre dépôt officiel.

Poussez les fichiers sur le dépôt distant.

Créer une nouvelle branche maBranche1 et publiez la sur le dépôt distant.

Dans maBranche1, modifier le fichier1, lui ajouter une virgule à la fin de chaque ligne. Ajouter une dernière ligne commençant par une étoile et comportant votre nom, prénom et terminant par un point.

Publier les modifications.

Créer une nouvelle branche maBranche2 et publiez là.

Dans maBranche2, modifier le fichier1, lui ajouter une virgule à la fin de chaque ligne. Ajouter une dernière ligne commençant par une étoile et comportant un nom et prénom différent par rapport aux modifications faites sur le même fichier de la branche1. Terminer par un point.

Utiliser *git diff* pour voir les modifications. Utiliser *git diff --tool-help* pour configurer un gui et l'utiliser.

Merge des branches secondaires puis merge avec le master

Merger la branche1 avec la branche 2, résoudre les éventuels conflits.

Merger la branche secondaire avec la branche principale, résoudre les éventuels conflits.

Supprimer les branches 1 et 2.

Merge des branches secondaires avec le master

Créer à nouveau 2 branches comme précédemment.

Y effectuer les mêmes modifications.

Merger le master avec la branche1. Puis merger le master avec la branche2.

Supprimer les branches 1 et 2.

Merge des branches secondaires et modifications en parallèle sur le master :

Créer à nouveau 2 branches comme précédemment.

Y effectuer les mêmes modifications.

Merger la branche1 avec la branche2.

Aller sur le master et modifier le fichier1. Y ajouter une virgule à la fin de chaque ligne. Ajouter une dernière ligne commençant par une étoile et comportant un nom et prénom différent par rapport aux modifications faites sur les mêmes fichiers de la branche1 et 2. Terminer par un point.

Puis merger le master avec la branche2.

Merge des branches secondaires et modifications en parallèle sur le master :

Créer à nouveau 2 branches comme précédemment.

Y effectuer les mêmes modifications mais avec un couple nom/prénom différent pour la branche2.

Aller sur le master et modifier le fichier1. Y ajouter une virgule à la fin de chaque ligne. Ajouter une dernière ligne commençant par une étoile et comportant un nom et prénom différent par rapport aux modifications faites sur les mêmes fichiers de la branche1 et 2. Terminer par un point.

Puis merger le master avec la branche1.

Modifier le fichier1 du master en supprimant des lignes, en ajoutant d'autres au milieu du fichier. Supprimer certaines virgules et ajouter des étoiles à certains début de ligne.

Merger le master et la branche2.

Utiliser mergetool:

Vous remettre dans l'un des cas de conflit précédent.

Expérimenter la commande *git mergetool* pour résoudre le conflit.

Merge avec theirs et ours:

Créer à nouveau 2 branches comme précédemment.

Y effectuer les mêmes modifications mais avec un couple nom/prénom différent pour la branche2.

Aller sur le master et modifier le fichier1. Y ajouter une virgule à la fin de chaque ligne. Ajouter une dernière ligne commençant par une étoile et comportant un nom et prénom différent par rapport aux modifications faites sur les mêmes fichiers de la branche1 et 2. Terminer par un point.

Puis merger le master avec la branche1 en conservant les fichiers de la branche1 (utiliser *git merge -X theirs*, cf : documentation git).

Modifier le fichier1 du master en supprimant des lignes, en ajoutant d'autres au milieu du fichier. Supprimer certaines virgules et ajouter des étoiles à certains début de ligne.

Merger le master et la branche2 en conservant les fichiers de la branche master (utiliser ours depuis la master).