

Git

Guide pour TIPE

MP2I Lycée Pierre de Fermat

Généralités

Git est un logiciel de gestion de versions, c'est à dire qu'il sert à stocker des fichiers et dossiers d'un projet, en conservant l'historique de toutes les modifications, de façon à ce que plusieurs personnes puissent participer au projet. Chaque personne possède sur son ordinateur une version locale du projet, et peut envoyer régulièrement sa version vers un serveur commun, ce qui a pour effet de fusionner les deux versions. Par exemple, si A travaille sur un fichier X.c, et B travaille sur un fichier Y.c, chacun de leur côté, lorsqu'ils envoient leurs modifications, la version du projet sur le serveur sera mise à jour avec les deux modifications. En fait, dans certains cas, vous pouvez même travailler sur le même fichier si vous touchez à des lignes différentes, mais c'est fortement déconseillé...

Github

Un projet géré avec Git doit être hébergé sur un serveur. Le plus simple est d'utiliser un site dédié, comme Github ou Gitlab. Dans ce guide, on utilisera Github. La première étape est donc de **vous créer un compte github**. Chaque membre du groupe doit créer son propre compte.

Windows

Si vous êtes sous Linux, vous pouvez directement passer à la partie suivante. Sous Windows, vous pouvez télécharger git (<https://git-scm.com/download/win>) pour obtenir une version avec une interface graphique ainsi qu'un petit terminal dédié à git. Une fois l'installation terminée, vous pouvez lancer l'interface graphique "Git GUI" ou bien le terminal "Git Bash".

1 SSH

Cette étape sert à lier votre ordinateur à votre compte github. Plus précisément, elle sert à convaincre Github que votre ordinateur est fiable et vous appartient bien. Pour cela, vous allez générer sur votre ordinateur un couple de clés SSH. Pour cela, tapez dans un terminal :

```
ssh-keygen
```

puis, appuyez sur Entrée jusqu'à la fin de la commande. Cela a pour effet de créer deux fichiers sur votre ordinateur : `"/home/[votre_ordi]/.ssh/id_rsa"` et `"/home/[votre_ordi]/.ssh/id_rsa.pub"`. Le premier est une clé privée, qui doit rester secrète, et le deuxième est la clé publique correspondante. Pour faire court, vous allez donner votre clé publique à Github, et vous pourrez ensuite communiquer avec le serveur de manière codée en utilisant le couple de clés. Pour accomplir cela, allez dans vos réglages Github (<https://github.com/settings/keys>) et ajoutez une nouvelle clé SSH. Donnez un titre à la clé, puis copiez le contenu du fichier `"/home/[votre_ordi]/.ssh/id_rsa.pub"`, c'est à dire votre clé **publique**, dans la zone correspondante.

2 Création du dépôt Git

Seule une personne doit suivre cette étape. Elle sert à créer sur votre compte github une page pour votre projet, et à faire les premières modifications sur le projet. Sur votre compte Github, créez un nouveau dépôt en cliquant sur “New”. Donnez-lui un nom, une description courte, puis cliquez sur “Create repository”. Vous êtes alors envoyés sur une page vous aidant à initialiser le projet. Cliquez sur l’option SSH, puis suivez les instructions commençant par “...or create a new repository on the command line”.

3 Ajout de membres

Une fois le dépôt créé, vous pouvez rajouter d’autres comptes Github au projet, pour leur donner les droits de modification. Pour cela, dans les réglages de votre projet, dans l’onglet “Collaborateurs”, vous pouvez ajouter des membres en leur envoyant une invitation via Github. La personne ajoutée doit accepter l’invitation, et une fois que c’est fait, il ne reste plus qu’à créer sur votre ordinateur une version locale du projet. Pour cela, allez sur la page principale du projet, et cliquez sur le bouton vert “Code”. Dans l’onglet SSH, copiez le texte de la forme “git@...”. Puis, lancez un terminal sur votre ordinateur, et, dans le dossier de votre TIPE, tapez les commandes suivantes :

```
git init
git clone [le texte copié]
```

4 Cycle d’utilisation Git

Git est un outil très complet, et donc complexe, mais il est possible de n’utiliser qu’un sous-ensemble des fonctionnalités proposées. Lorsque vous utilisez Git via un terminal, toutes les commandes commenceront par `git`, et devront être utilisées dans le dossier où votre copie locale du projet est stockée. Les 4 commandes vitales sont :

- `git pull` : sert à récupérer la dernière version du projet, avec toutes les modifications effectuées par les autres membres
- `git add [fichier1] [fichier2] ... [fichierN]` : sert à rajouter des fichiers au projet. Le simple fait de créer un fichier (par exemple avec `touch`) dans le dossier ne suffit pas, il faut aussi utiliser `git add` pour signaler l’ajout du fichier au projet.
- `git commit -a -m "message informatif"` : sert à emballer les modifications que vous avez effectué localement. Vous êtes encouragés à faire des commits réguliers, car en cas de soucis ce sont les commits qui vous permettent de récupérer des anciennes versions. Mettez des messages explicites (ajout de fichiers pour faire X, correction du bug Y, début de la partie sur Z) pour que les autres membres puissent voir rapidement ce que vous avez fait.
- `git push` : sert à envoyer sur le serveur votre version du projet, et à appliquer les modifications nécessaires pour mettre la version distante à jour.

Ainsi, une session de travail avec git ressemble généralement à :

1. Git pull la version la plus à jour du projet
2. Travailler sur la version locale, en faisant des git commits régulier (et en n’oubliant pas de git add les nouveaux fichiers).
3. Git push à la fin de la session pour mettre à jour la version distante du projet.