

Travailler à plusieurs sur GitHub

Jusque là nous avons travaillé avec un dépôt (ou encore “repository”) local sur Git et GitHub, et sur un dépôt distant sur GitHub. Mais comment communiquer en local avec un dépôt qui est sur internet, comme celui que nous avons créé sur GitHub?

Ce TP est l’occasion d’introduire la notion de **remote** git.

Evidemment un “remote” n’est pas propre à GitHub, il existe d’autres plateformes d’hébergement de projets git telles que BitBucket ou encore GitLab. Vous pouvez même créer vos propres serveurs Git et les rendre accessibles sur le réseau, ce que font beaucoup d’entreprises.

Travailler sur un dépôt distant présente deux avantages:

- Pouvoir travailler à plusieurs sur un même projet;
- Ne pas perdre son travail si on a un problème en local;

Revoir les commandes principales de Git

- Accéder à ce tutoriel rapide <https://try.github.io> et le terminer.
- Effectuer une capture d’écran.

Configuration des remotes

Un remote représente le chemin vers un dépôt distant. Un projet git peut avoir plusieurs remote identifiés par des noms.

Par exemple, le remote “origin” pourrait rediriger vers le dépôt principal d’un projet sur GitHub quand le remote “mickael” pourrait rediriger vers le fork (la copie donc) de ce projet Github (<https://github.com/mickael/le-projet.git>).

Collaborer sur GitHub en mode CLI

Récupérer un dépôt distant (sur GitHub)

Quand on souhaite récupérer un projet sur GitHub, on effectue un clone du dépôt en local.

- Ouvrir un terminal et exécuter la commande suivante:

```
git clone https://github.com/mickaelandrieu/magnificent-website.git
```

- Que s'est-il passé?
- Exécuter les commandes `git remote` puis `git remote -v`.
- Accéder à GitHub, se connecter et forker le dépôt (Accessible sur <https://github.com/mickaelandrieu/magnificent-website>).
- Dans le terminal, exécuter la commande `git remote add mon-remote https://github.com/votre-login/magnificent-website`.
- Exécuter les commandes `git remote` puis `git remote -v`.
- Que s'est-il passé?

Si vous voulez en savoir plus sur les remotes, se référer à la documentation officielle: <https://git-scm.com/book/fr/v1/Les-bases-de-Git-Travailler-avec-des-d%C3%A9p%C3%B4ts-distants>.

Envoyer du code vers un dépôt distant (sur GitHub)

Le but ici est d'effectuer une modification sur le fork du dépôt "magnificent-website" en local et de la rendre disponible sur Github.

- Dans le dossier "magnificent-website", accéder au fichier "[README.md](#)".
- Supprimer tout le contenu et le remplacer par le texte suivant "# This is an update from local".
- Ajouter et commiter la modification avec le message suivant: "first commit available on GitHub".
- Exécuter la commande suivante: `git push mon-remote master` et donner les informations d'identification demandées par le terminal.
- Accéder à votre projet sur GitHub (Accessible sur <https://github.com/votre-login/magnificent-website>).
- Effectuer une capture d'écran.
- Que s'est-il passé?

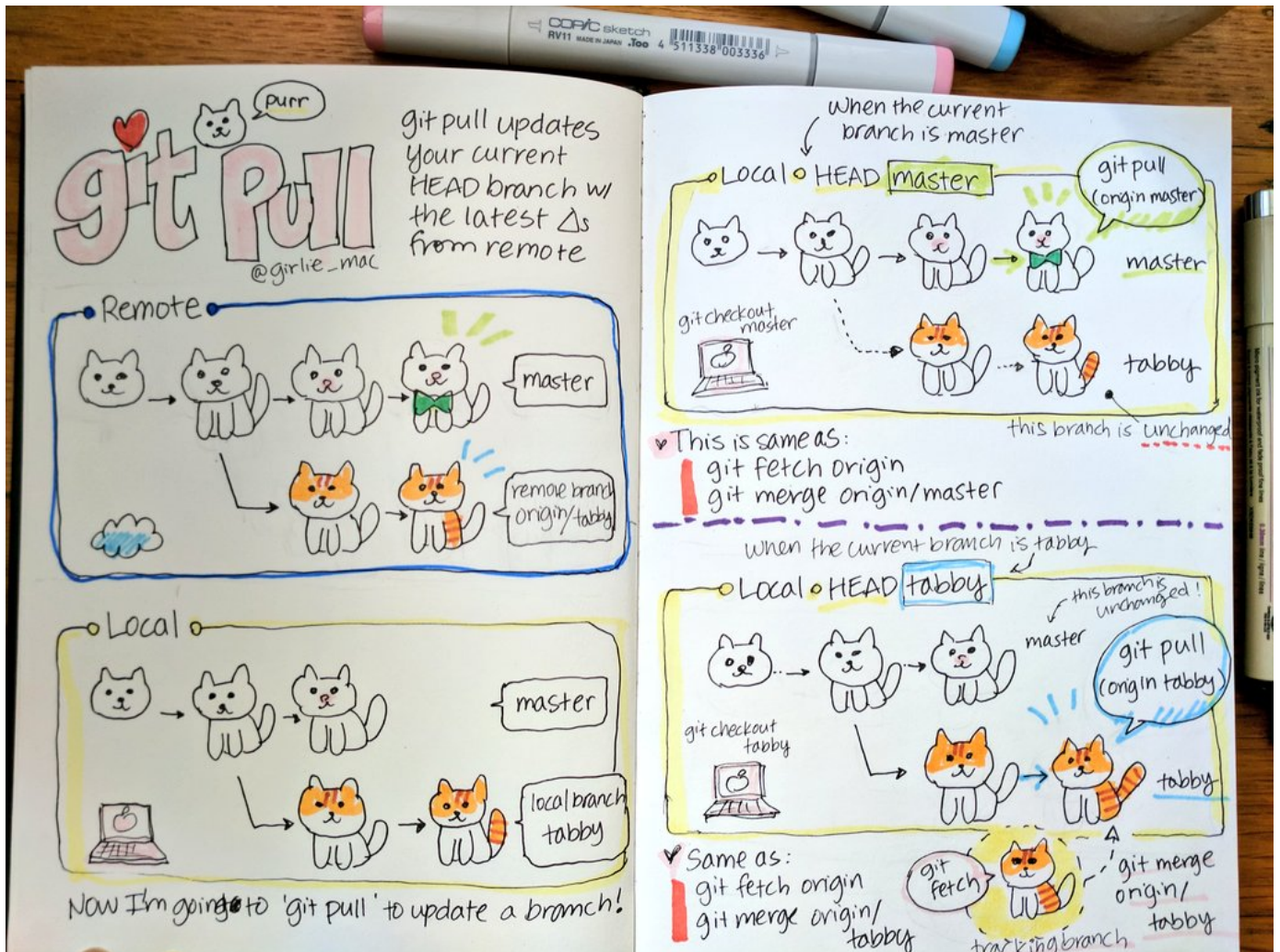
Récupérer du code vers un dépôt distant (à partir de GitHub)

Le but ici est de synchroniser le dépôt local avec la version distante sur GitHub.

- Dans GitHub et sur votre dépôt, re-modifier le fichier [README.md](#) et ajouter le contenu suivant à la ligne 2: "## Edité à partir de GitHub".
- Commiter avec le message de commit suivant: "second commit created in GitHub").

- Dans le terminal et dans le dossier “magnificent-website” exécuter la commande `git pull mon-remote master`.
- Exécuter la commande `git log` et effectuer une capture d’écran de la fenêtre du terminal.
- Que s’est-il passé?

Ce schéma peut vous aider à comprendre ce qu’il se passe durant un “git pull”.



Si on souhaite récupérer l'historique git d'une branche sans récupérer le code correspondant, il faut utiliser la commande “fetch”.

Collaborer sur GitHub avec GitKraken

Récupérer un dépôt distant (sur GitHub)

- Supprimer le dossier “magnificent-website” de votre ordinateur.
- Ouvrir le logiciel “GitKraKen” et sélectionner l’option “Clone” dans le menu “File”.

- Dans la fenêtre qui s’affiche, choisir l’option GitHub et suivre le processus pour s’authentifier sur GitHub.
- Choisir le dépôt à cloner (“magnificent-website”) et l’endroit où le dossier sera créé.
- Cliquer sur le bouton “Clone the repo!”.
- Cliquer sur le bouton “Open now” dans la notification qui est apparu en haut.
- Prendre une capture d’écran.

Envoyer du code vers un dépôt distant (sur GitHub)

- Dans le dossier “magnificent-website”, accéder au fichier “[README.md](#)”
- Supprimer tout le contenu et le remplacer par le texte suivant “# This is an update from local with GitKraken”.
- Ajouter et commiter la modification avec le message suivant: “second commit available on GitHub using GitKraken”.
- Cliquer sur le bouton “Push” dans le menu.

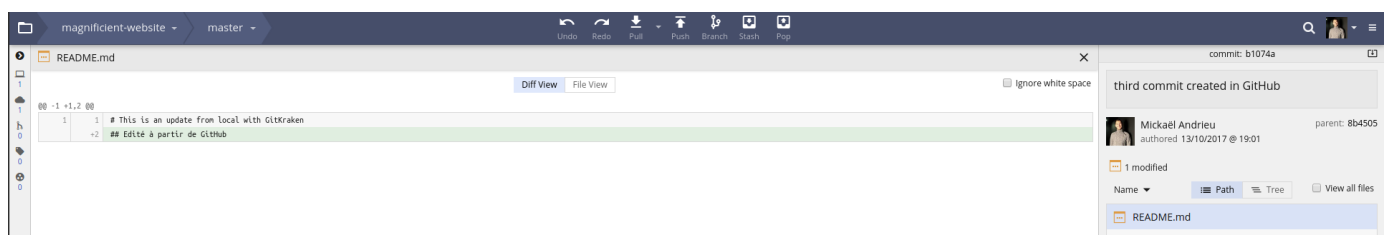
Comprendre l’authentification sur GitHub

GitKraken n’a pas demandé de fournir les identifiants pour pousser du code sur GitHub. C’est parce ce logiciel s’est authentifié en SSH (Secure Shell) alors que le remote git était configuré en HTTPS. Pour résumer, GitHub gère 2 types d’authentification:

- **HTTPS** (remote en <https://github.com/xxxx>) qui nécessite de mettre ses identifiants;
- **SSH** (remote en <git@github.com/xxxx>) qui nécessite la création d’une clé SSH, ce qu’a fait GitKraken;

Récupérer du code vers un dépôt distant (à partir de GitHub)

- Dans GitHub et sur votre dépôt, re-modifier le fichier [README.md](#) et ajouter le contenu suivant à la ligne 2: “## Edité à partir de GitHub”.
- Commiter avec le message de commit suivant: “third commit created in GitHub”).
- Dans GitKraken, appuyer sur le bouton “Pull”: le commit devrait apparaître dans l’historique.
- Prendre une capture d’écran.



Rendus du TP

- Sur le dépôt <https://github.com/mickaelandrieu/magnificent-website>, créer une issue “TP Git/GitHub” et uploader l’ensemble des captures d’écran réalisées durant ce TP.