Git & GitHub - Partie 3 Activité

TRAVAUX

Expliquer pour une personne qui connaît le développement web, mais n'a jamais utilisé Git les notions suivantes :

1. Qu'est-ce qu'un commit;
2. À quoi sert la commande git log;
3. Qu'est-ce qu'une branche.

PREAMBULE

Git est un logiciel de gestion de versions décentralisé ou DVCS (Distributed Version Control System) utilisé pour les projets de développements open source ou commerciaux.

Il peut être installé dans différent environnement et notamment Windows 10. Une fois installé le logiciel est accessible de différentes manières :

- Git CMD est identique à la ligne de commande habituelle Windows et permet d'exécuter les fonctionnalités de Git au travers de la ligne de commande Windows.

- Git Bash émule un environnement Bash sous Windows sous forme de ligne de commande. Git Bash permet d'exécuter toutes les commandes Git et la plupart des commandes Unix.

- Enfin, Git GUI est une interface graphique qui vous permet d'utiliser Git sans la ligne de commande.

Un « repository », ou « Projet Git », regroupe l'ensemble des fichiers et/ou dossier associé au projet ainsi que l'historique de l'ensemble des modifications effectuées. L'historique des modifications est constitué d'un ensemble d'instantané appelés « commit ». L'ensemble des « commit » sont accessibles grâce à la commande « git log ». Enfin les développements peuvent être organiser sous forme de chantier, travaillant en parallèle, grâce au concept de « branche ».

Git étant un DVCS, les « repository » permettent à toute personne ayant une copie du « repository » d'accéder à l'ensemble des éléments et à leur historique.

Travailler avec des "repository" permet d'organiser et de sécuriser les projets de développements Les développeurs peuvent corriger des bugs, développer des nouvelles fonctionnalités, sans altérer les axes majeurs de développements grâce à la notion de « branche ».

Au travers de plateforme comme GitHub, Git offres des fonctionnalités pour encore plus de transparence et de collaboration.

En effet, GitHub est une plateforme sur le web qui facilite la gestion du code grâce à des fonctionnalités de travail collaboratif et de gestion de version. Elle permet de travailler à plusieurs sur des projets quel que soit la localisation de l'équipe. L'ensemble des ressources d'un projet peuvent être gérées (code, fichier texte, librairie, photo,….).

1/ QU'EST-CE QU'UN COMMIT ?

Un « commit » permet de sauvegarder les modifications dans le « repository » local.

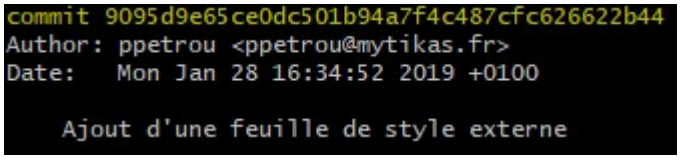
Il est important de noter qu'il faut préciser à Git les modifications que l'on souhaite inclure dans le « commit » avant de lancer la commande « git commit ». Un fichier ne sera pas automatiquement inclus dans le « commit » à venir simplement parce qu'il a été modifié. Il faut au préalable utiliser la commande « git add » afin de marquer les modifications que l'on souhaite prendre de compte.

Il est important de noter que dans Git, un « commit » sauvegarde le nouveau « commit » dans la « repository » local de Git. L'échange de « commit » avec une plateforme comme GitHub ne pourra se faire qu'au travers des commandes réalisées manuellement du type : « git pull », « git push » …

Option Importante de la commande « git commit » :

- -m <message> permet de saisir le message d'information (à mettre entre guillemets) qui doit être concis et précis et permettre aux participants du projets de comprendre les modifications.
- -a permet d'inclure tous les fichiers en cours de modification dans ce « commit ». Les nouveaux fichiers non tracés ne seront pas inclus.

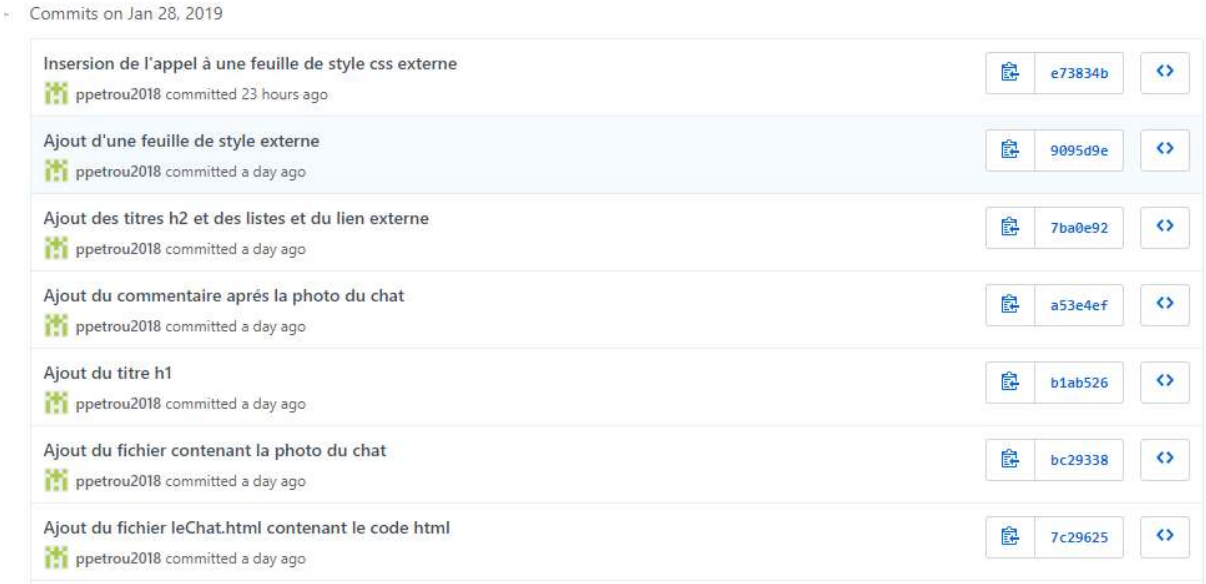
Un exemple ci-dessous de la console Git Bash :



Chaque « commit » contient :

- Le numéro unique sha du « commit »
- L'auteur
- La date et l'heure
- Le message permettant d'expliciter la modification effectuée

L'image ci-dessous permet de voir l'écran GitHub après envoi des modifications sur la plateforme.



En cliquant sur le « commit » correspondant on obtient l'écran suivant :

On constate que grâce à cet outil de versionning il est possible de voir en vert le détail des modifications. Dans ce cas précis l'ensemble des lignes ont été créés avec la constitution du fichier chat.css.

2/ A QUOI SERT LA COMMANDE git log ?

La commande « git log » permet de voir l'historique de l'ensemble des « commits ». Les logs de Git permettent de revoir et lire l'ensemble de ce qui s'est passé dans le « repository » correspondant. L'historique est accessible avec l'instruction "git log". Cette commande possède de nombreuses options pour afficher l'historique des « commits ».

Le résultat de commande git log contient les éléments suivants pour chaque « commit » :

- Un commit hash (SHA1 : 40 caractères). Il est unique car généré sur le contenu du « commit » lui-même.

- Le nom et l'email de la personne qui a effectué le « commit »
- Le message du « commit »

A titre d'exemple ci-dessous un exemple de log.

```
$ git log
commit e73834bbe75de7b6e07257b95c5a67755a82b06e (HEAD -> master, origin/master,
origin/HEAD, une-autre, mon-test)
Author: ppetrou <ppetrou@mytikas.fr>
Date:   Mon Jan 28 16:36:11 2019 +0100

    Insersion de l'appel à une feuille de style css externe

commit 9095d9e65ce0dc501b94a7f4c487cfc626622b44
Author: ppetrou <ppetrou@mytikas.fr>
Date:   Mon Jan 28 16:34:52 2019 +0100

    Ajout d'une feuille de style externe

commit 7ba0e922d2cc56887af3c1dfa1415ca8fbb18599
Author: ppetrou <ppetrou@mytikas.fr>
Date:   Mon Jan 28 16:19:29 2019 +0100

    Ajout des titres h2 et des listes et du lien externe

commit a53e4ef7c018bc89e8da5248d97ca652349922a5
Author: ppetrou <ppetrou@mytikas.fr>
Date:   Mon Jan 28 16:03:04 2019 +0100

    Ajout du commentaire aprés la photo du chat

commit b1ab52652d90e4b279c6c7b4c9b804ece2b329af
Author: ppetrou <ppetrou@mytikas.fr>
Date:   Mon Jan 28 15:57:33 2019 +0100

    Ajout du titre h1

commit bc29338c3d0a09052641f6726c38d93ecb67cb9d
Author: ppetrou <ppetrou@mytikas.fr>
Date:   Mon Jan 28 15:47:24 2019 +0100

    Ajout du fichier contenant la photo du chat

commit 7c29625bdb728dd31e21041c68cbfe88f48e55ea
Author: ppetrou <ppetrou@mytikas.fr>
Date:   Mon Jan 28 15:45:44 2019 +0100

    Ajout du fichier leChat.html contenant le code html

commit 7ba5396e7b9af0504c85a53cbdcd85270f00d310
Author: ppetrou2018 <45563190+ppetrou2018@users.noreply.github.com>
Date:   Mon Jan 28 15:07:55 2019 +0100

    Initial commit
```

3/ QU'EST-CE Q'UNE « BRANCH » ?

Une « branch » dans Git est simplement un pointeur sur des « commit ». Par défaut, la branche principale est appelée « master ».

A titre d'exemple, ci-dessous les différentes « branch » grâce à la commande « git branch » :

La « branch » en vert est la branche principale. D'autres branches ont été créés pour la compréhension.

La commande « git checkout » suivi du nom de la branche dans laquelle nous voulons nous positionnez permet de se déplacer dans les branches :

CONCLUSION

Vous pouvez consulter les nombreuses aides en ligne sur les différents sites pour aller plus loin.

Pour aller plus loin et mieux comprendre les différents concepts je vous conseille de vous rendre sur le site : https://git-scm.com/about et de télécharger l'ebook qui est disponible.

Sur le site le chapitre concernant les branches et le déplacement des pointeurs est très bien fait.

Et un grand merci à Open Class Room.

****