GIT

Découvrir les forks et pull :

Un pull request, c’est le fait de proposer sur un dépôt ou nous n’avons pas le droit de d’ecriture, des modifications. Ça se passe en plusieurs étapes : la première ça va être de dupliquer le projet via un fork, on se retrouve alors avec une copie de projet, mais cette fois si vue que la partie nous appartient nous pouvons réaliser des commit et modifier ce projet, et une fois le projet est modifié, on envoie un Pull request au propriétaire du dépôt initial afin de lui proposer nos modifications, cette personne va pouvoir accepter ou non les modifications pour les intégrés dans son propre dépôt.

Un Pull request constitué d’un titre et d’un Id qui permet de l’identifier, l’état (ouvert ou fermé), le nom de la personne qui la crée ainsi que des informations sur ses actions.

Pour réaliser un Pull request sur un projet, il faut d’abord réaliser un fork de celui si, comme ça nous aurons accès en écriture à cette copie de dépôt.

Utiliser un dépôt distant :

Comprendre le modèle distribué :

C’est mieux de travailler avec un dépôt distant pour plusieurs raisons :

- la première, est d’avoir une sauvegarde de notre projet ailleurs que sur notre PC

- la deuxième, c’est de pouvoir partagé ce dépôt avec d’autre personnes afin de travailler ensemble sur le même projet.

Il y a deux grandes familles de modèles de gestion de versions :

\* Modèle centralisé : (CVS, SVN) la particularité de ce modèle, c’est qu’il utilise un serveur central, tous les développeurs vont se connecter à ce serveur afin de travailler sur le projet, mais ils vont pouvoir récupérer uniquement une version à la foi du projet.

\* Modèle distribué : (Git, Mercurial) la différence de ce modèle, c’est que cette fois si, l’intégralité de dépôt va être copié sur les ordinateurs de développeurs et pas seulement une version donnée du projet, alors l’intérêt prôniez c’est que les personnes n’en pas besoin d’être en permanence connectés à ce serveur pour travailler sur le projet, en effet l’intégralité du projet va être copié sur leur machine.

{Image}

Etant donné, notre dépôt Git qui est hébergé sur un serveur et comporte un ensemble de commit, lorsque les développeurs veulent travailler sur ce dépôt distant, il vont fait ce qu’on appelle un dupliquât de ce dépôt, cloner totalement le déport sur leur machines, Git va copier l’intégralité de ce dépôt Git sur les dépôts locaux des développeurs, si par exemple le dépôt du développeur A décidé de faire une modification en local sur son dépôt, alors on va alors utilisé le commit, envoie l’intégralité des commit supplémentaires qui la réalisés sur le dépôt distant . A ce moment le développeur qui possède le dépôt local B va pouvoir se synchroniser également avec ce dépôt distant mais cette fois si pour récupérer les modifications, ainsi nous allons nous retrouver avec les modifications faites par le développeur A qui sont présent également sur le dépôt distant, et donc chez tous les autres développeurs il faut qui se seront synchroniser avec le dépôt.

Cloner un dépôt Git :

Après la création du dépôt Git sur Github, il faut le récupérer, cloner, en local sur notre machine.

{image }

Gérer les remotes :

Pour relier notre dépôt distant avec notre dépôt local qui contient les modifications que nous avons fait depuis le début. Pour cela nous devons modifier les remotes de notre dépôt local.

Un remote est tout simplement un pointeur vers un autre dépôt qui va nous servir de référence.