

Git & GitHub



VERSIONNING



CLOUD

C'est quoi Git ?



C'est un logiciel de versionning. Il permet de garder une trace de toutes les modifications apportées à un code, un fichier ou autres.

C'est quoi GitHub



GitHub permet de travailler à plusieurs sur le même code. Il permet donc de créer un cloud où toutes les personnes pourront travailler en conservant les traces de chaque modifications.

Repository

Un repository ou repo est un simplement un dossier où vous pourrez y stocker tous vos dossiers et documents.

Un repository possède un README.md qui vous permettra d'expliquer votre projet.

Commits

Il existe 4 commandes principales, commits est la première. Elle permet de sauvegarder la version sur laquelle on travaille. Cette version va s'ajouter aux anciennes et on pourra alors constater tous les changements effectués avec la dernière en date.

Pull

Git est un logiciel où les changements sont faits en local (sur notre PC). Pour récupérer le repository entier de GitHub sur notre PC afin d'y travailler dessus, nous allons devoir utiliser la commande pull.

Push

Une fois quelques commits (changements) effectués sur notre fichier en local, nous allons pouvoir envoyer notre travail sur GitHub avec la commande Push.

Merge

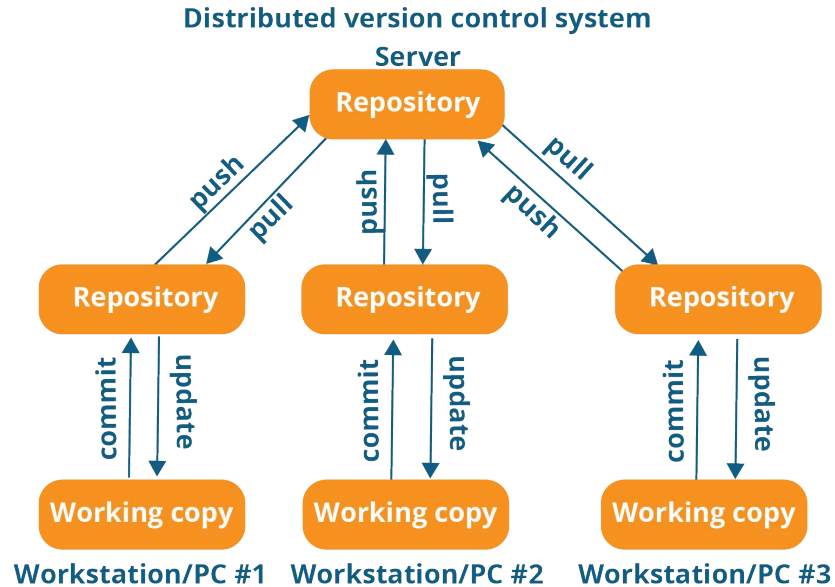
Cependant, si plusieurs personnes travaillent sur le même projet et qu'il "push" en même temps. Nous allons devoir fusionner ces deux versions. C'est là qu'intervient la commande merge.

Branches

Les branches permettent d'organiser notre GitHub en instaurant une hiérarchie et des protocoles entre tous les utilisateurs. Tout le monde ne pourra pas push et modifier tout le code d'un coup, avec le système de branche, il passera par des vérifications d'autres personnes avant d'être validé.



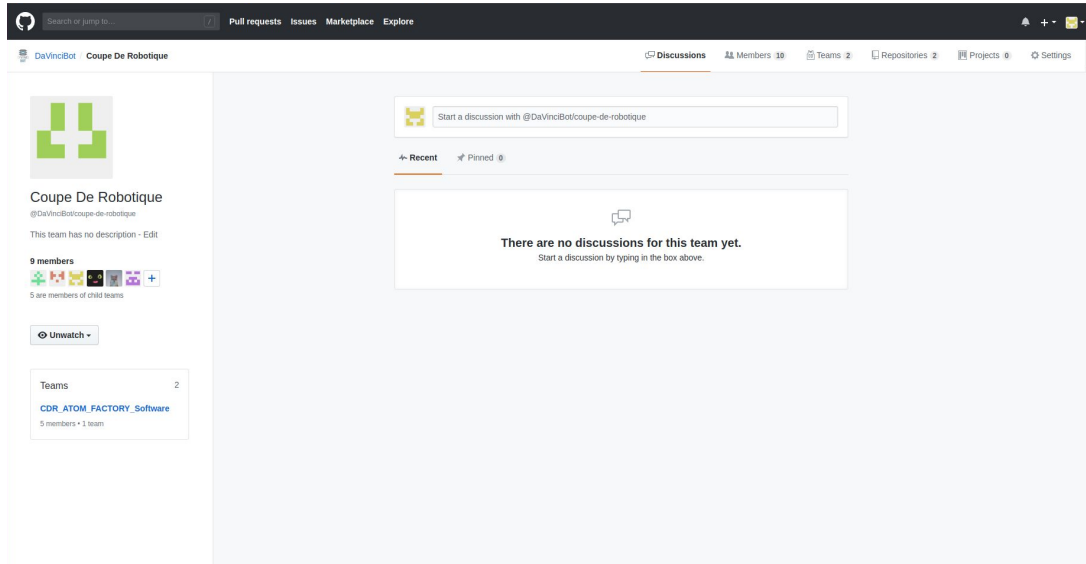
GitHub - Fonctionnement



L'organisation d'un GitHub

Depuis peu, nous pouvons avoir une vraie organisation de notre GitHub. Nous pouvons par exemple créer différentes “Teams” qui pourront travailler sur des projets différents et donc avoir un canal de discussion, une partie du GitHub différent. Ce système s'apparente aux logiciels déjà existants tels que Teams ou Slack.

GitHub - TEAMS



Sur GitHub, il est possible de faire des équipes. Les équipes sont liées à des projets (nous allons en parler dans le slide suivant)
Chaque équipes ont des repo associés.

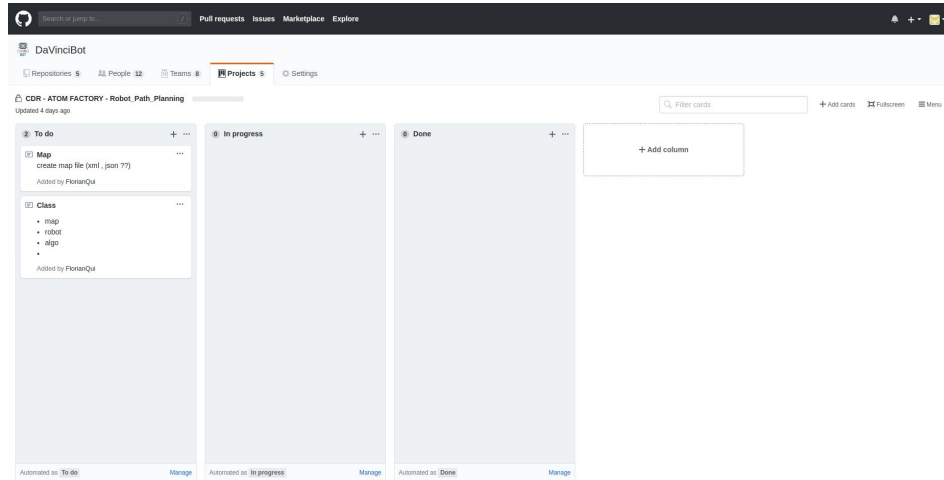
Sur la photo, on peut voir l'équipe "Coupe de Robotique", il y a 10 membres, 2 sous teams, 2 repos, et 0 projets.

Les équipes imbriquées permettent de bien organiser le projet.

Les équipes sont les seuls à pouvoir apporter des modifications à son repo lié

Nous reviendrons plus en détail sur la façon de faire les modifications.

GitHub - PROJECTS



Nous allons parler des Projects de GitHub
C'est un système de management projet,
très proche de Trello.

Cela permet d'organiser une ToDo List.

Les projets peuvent appartenir à des
équipes.

GitHub - PULL REQUEST

Nous travaillons sur un outil collaboratif, cela veut dire que tout le monde n'a pas accès à l'ensemble des ressources fournies par GitHub.

Il existe trois type,

- Owner (les membres du bureau)
- Admin (les chefs de projets)
- Member (l'équipe projet)

Afin de travailler dans un environnement propre, la branche master a été restreinte.

Pour rendre votre travail sur disponible sur la branche master, qui je rappelle est la branche finale du repo, il est impératif de faire des modifications dans une branche qui vous aurez préalablement créée. Un fois ces modifications réalisées, il est possible de les rendre visible sur la branche master en effectuant un PULL REQUEST.

GitHub - PULL REQUEST

Un PULL REQUEST permet la relecture de vos modifications par d'autres membres de l'équipe. Plusieurs personnes peuvent approuver celles-ci. Néanmoins, seul un "Admin" peut merge votre branche vers la branche master.

Nous rappelons aussi que le nombre de branches possible est infini.