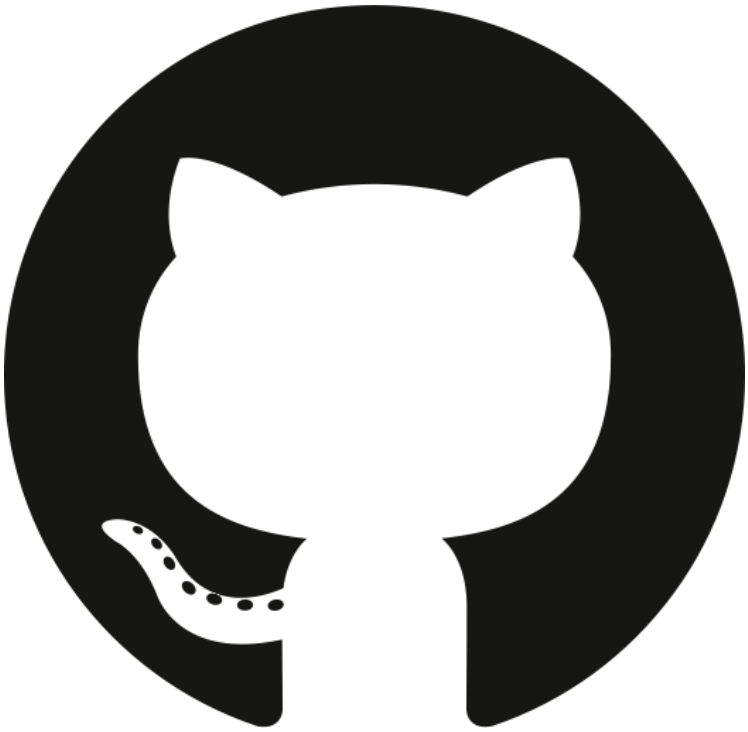
VEILLE TECHNOLOGIQUE

GITHUB



Sommaire

Introduction

GitHub est un service web d'hébergement pour le développement de logiciels lancé en 2008 par Tom Preston-Werner, Chris Wanstrath et PJ Hyett. Il a pour but d'aider les développeurs à stocker et à gérer leur code, ainsi qu'à suivre et contrôler les modifications qui lui sont apportées.

Pour comprendre exactement ce qu'est GitHub, il faut comprendre ce qu'est Git, mais aussi comprendre ce pour quoi le site est devenu aussi populaire, accumulant plus de 73 millions d'utilisateurs en 2021.

I. Le Git de GitHub

A. Git

Git est un logiciel de versioning open-source créé en 2005 par Linus Torvalds, le créateur de Linux.

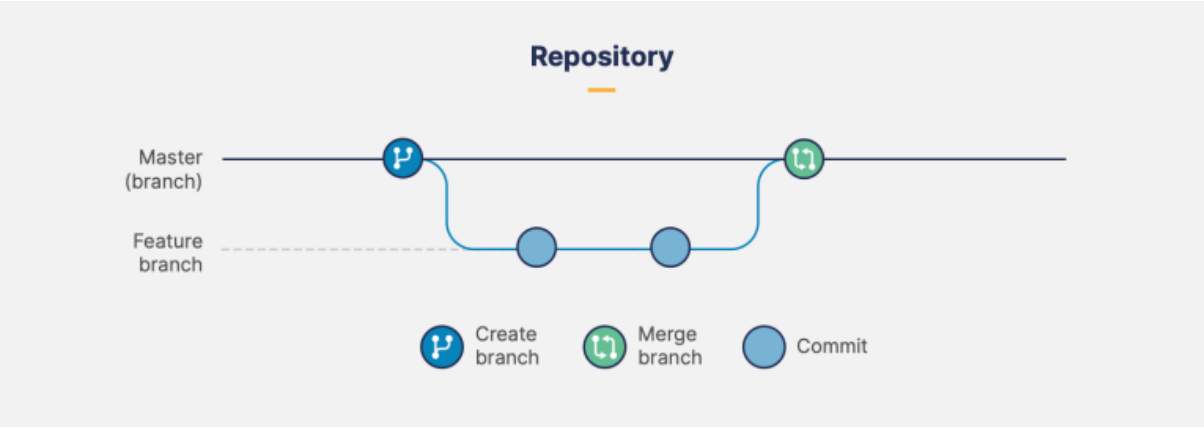
D'après un sondage auprès des développeurs de Stack Overflow, plus de 87% des développeurs utilisent Git, on peut donc facilement dire qu'il s'agit du logiciel de versioning le plus populaire, avec plus de 200 millions de dépôts Git en 2021.

B. Qu'est-ce que le contrôle de version ?

Un logiciel de contrôle de version va permettre de conserver un historique des modifications effectuées sur un projet afin de pouvoir rapidement identifier les changements effectués et de revenir à une ancienne version en cas de problème.

Ces logiciels sont aujourd'hui quasiment incontournables car ils facilitent grandement la gestion de projets et permettent de travailler en équipe de manière beaucoup plus efficace.

Si on voulait tester une fonctionnalité sans Git, on devrait faire une copie du projet au cas où il faudrait le restaurer. Pour une personne, ça ne serait pas un problème majeur, mais à partir d'une équipe de plusieurs personnes, il devient compliqué de se partager toutes les modifications de manière synchronisée, et surtout de retenir quelles lignes des fichiers ont été modifiées. Au lieu de cela, le contrôle de version permet aux développeurs de travailler en toute sécurité à travers les branchements et les fusions :



À l'aide d'une branche, le développeur va dupliquer une partie du dépôt, où il pourra alors apporter des modifications en toute sécurité à cette partie du code sans affecter le reste du projet : commit.

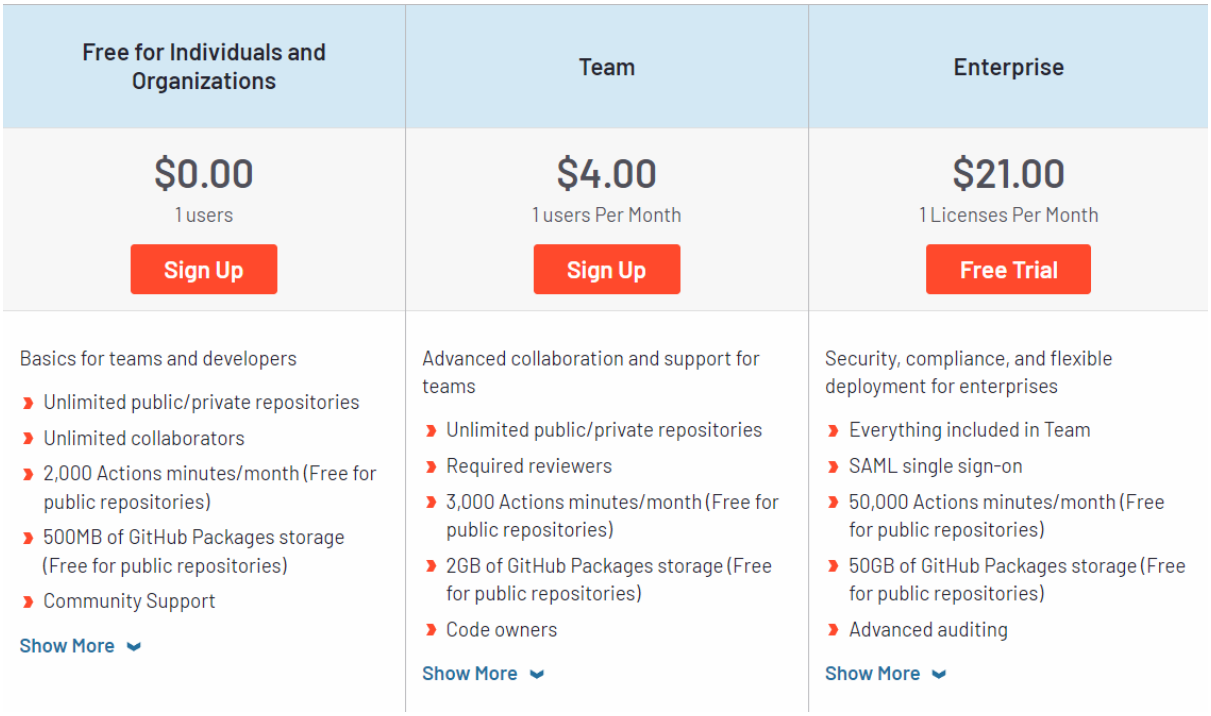
Ensuite, une fois qu'il aura réussi à faire fonctionner correctement sa partie du code, il pourra fusionner ce code avec le code source principal pour le rendre officiel : merge branch.

Tous les changements sont ensuite suivis, que ce soit la création ou la suppression d'un fichier, la modification de son contenu, et peuvent être annulés si nécessaire.

II. Le Hub de GitHub

A. Alors à quoi sert GitHub ?

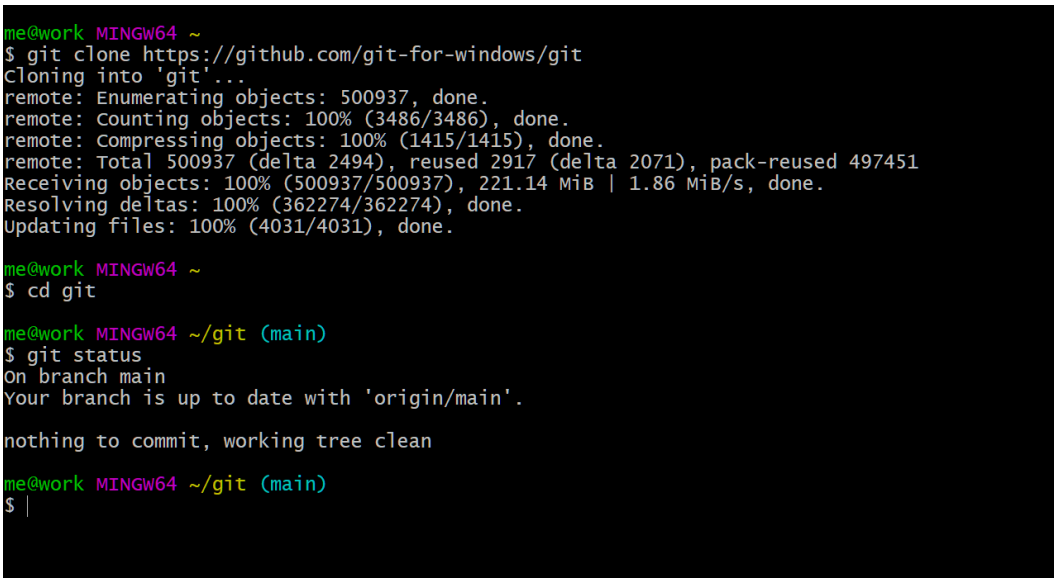
GitHub va nous permettre d'utiliser ce système de gestion de version sans passer par les lignes de commande de Git. On va avoir une interface graphique par laquelle on aura accès à des fonctionnalités, limitées selon le type de compte :



Free for Individuals and Organizations	Team	Enterprise
$0.00 1 users Sign Up	$4.00 1 users Per Month Sign Up	$21.00 1 Licenses Per Month Free Trial
Basics for teams and developers Unlimited public/private repositories Unlimited collaborators 2,000 Actions minutes/month (Free for public repositories) 500MB of GitHub Packages storage (Free for public repositories) Community Support Show More	Advanced collaboration and support for teams Unlimited public/private repositories Required reviewers 3,000 Actions minutes/month (Free for public repositories) 2GB of GitHub Packages storage (Free for public repositories) Code owners Show More	Security, compliance, and flexible deployment for enterprises Everything included in Team SAML single sign-on 50,000 Actions minutes/month (Free for public repositories) 50GB of GitHub Packages storage (Free for public repositories) Advanced auditing Show More

- Compte gratuit : on peut créer des dépôts publics et privés de manière illimitée, 500mb de stockage, et des fonctionnalités comme les branches protégées ou mettre en ligne mais seulement sur les dépôts publics
- Comptes Pro à partir de 4 dollars par mois qui nous donneront + de stockage, et nous permettront d'utiliser toutes les fonctionnalités du compte gratuit mais aussi sur des dépôts privés + GitHub Codespaces (= run le dépôt contrairement à GitHub.dev qui ne permet que de modifier / créer des fichiers et commit)

L'intérêt de GitHub dans son interface est qu'elle est suffisamment conviviale pour que même les codeurs débutants puissent profiter de Git. Sans GitHub, l'utilisation de Git nécessite généralement un peu plus de connaissances techniques et l'utilisation de lignes de commande.



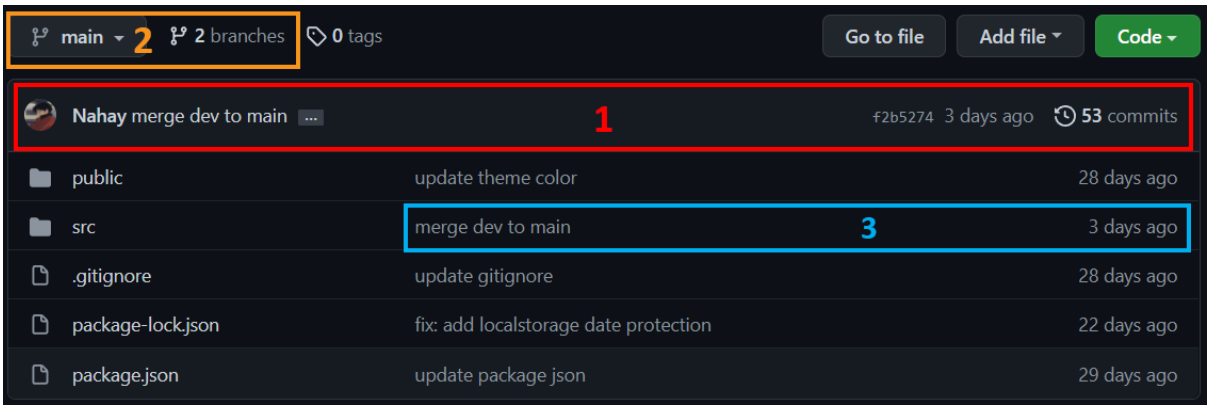
Logiciel Git Bash permettant l'utilisation de Git en lignes de commande

Et d'ailleurs, il n'y a pas forcément que les programmeurs qui vont utiliser GitHub. Le système de versions peut intéresser des auteurs de livre, par exemple, même s'il existe pour cela de meilleures alternatives.

N'importe qui peut donc s'inscrire et héberger gratuitement un dépôt de code public, ce qui rend GitHub particulièrement populaire auprès des projets open-source.

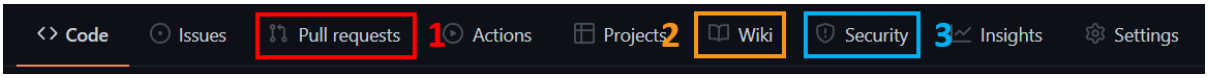
B. La gestion d'un référentiel via GitHub

Voilà à quoi ressemble l'interface graphique d'un dépôt : on va avoir l'arborescence de tous les dossiers et fichiers sous leur dernière version publiée.



- [1] On peut voir le dernier commit publié, avec le nom de celui qui l'a publié, le nom et la date du commit. À droite, on peut aussi voir le nombre total de commit qui a été réalisé sur le dépôt. Si on clique dessus on aura la liste des tous les commit présentés de la même façon que celui-ci.
- [2] Ici, on va pouvoir voir les différentes branches qui ont été créées par différents collaborateurs ou pour des versions différentes (branche dev et branche prod par exemple).
- [3] Pour chaque fichier, on peut voir le nom et la date du commit qui contient leur version la plus récente.

GitHub organise d'une certaine façon un journal des modifications, puisqu'on pourra voir qui a modifié quoi, quand et où ces fichiers sont stockés.

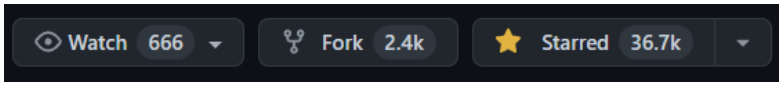


- [1] Ici on va avoir la liste des pull request *(= cela consiste à demander à la personne responsable d'inclure votre code*) par des développeurs externes.

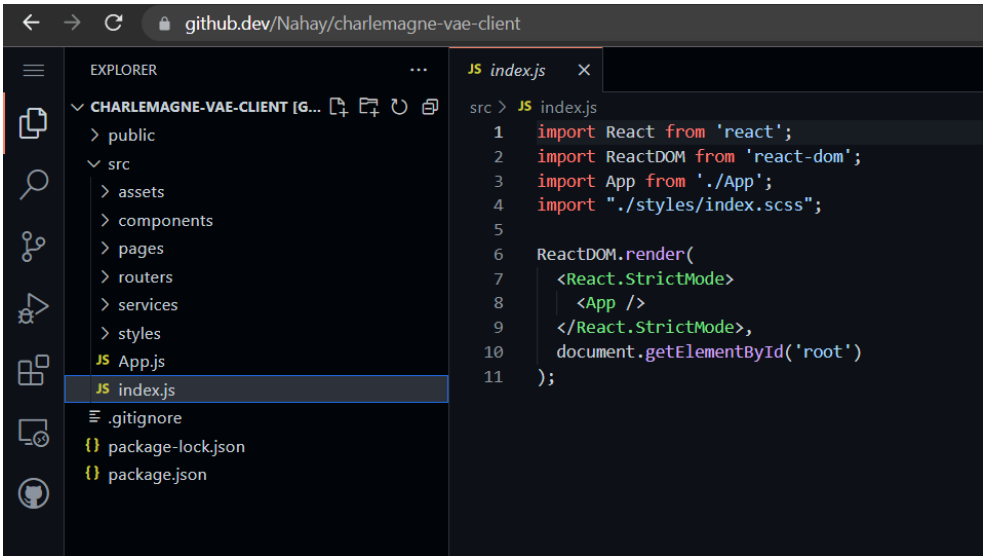
- [2] On pourra créer un wiki pour notre projet. Il s'agit d'un ensemble de fichier markdown qui permettent de créer une documentation.

- [3] Il y a aussi un onglet sécurité où on aura des alertes, par exemple si notre projet contient des librairies qui présentent des failles de sécurité. GitHub met à disposition un bot qui va s'occuper de mettre à jour les librairies à une version stable.

La fonctionnalité qui a le plus mis GitHub en avant est le Fork : vous allez créer une copie du dépôt dans vos propres dépôts. Le but est de créer ses modifications et de soumettre ce code en faisant un pull request. Cela va notamment encourager le développement ultérieur de projets et forcément, encourage le côté social qu'instaure GitHub.



La particularité du site est que l'on peut à la fois modifier le code d'un fichier en cliquant dessus dans l'arborescence mais aussi via l'éditeur GitHub Dev : on y accède soit en changeant simplement le .com par .dev, ou en appuyant sur la touche « point » . Cet éditeur ne permet cependant que de parcourir et modifier ses fichiers, commit, pull, mais pas de run ou debug : pour cela il faudra devenir Pro afin d'acquérir GitHub CodeSpaces.

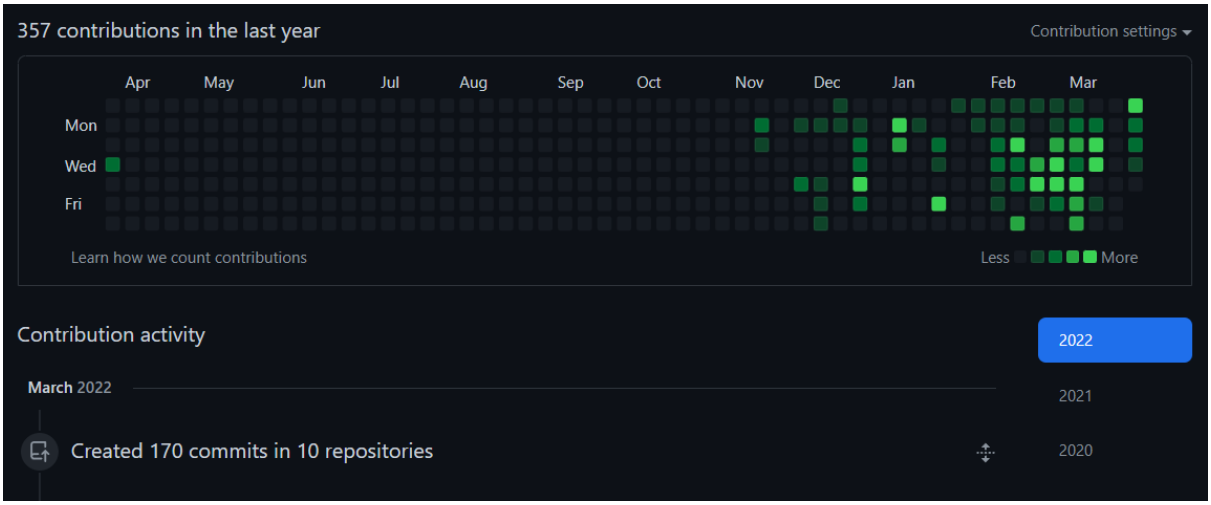


Aperçu de l'éditeur en ligne GitHub dev

C. L’aspect communautaire de GitHub

Comme dit précédemment avec le fork, GitHub est centré vers l'aspect social du développement.

Chaque utilisateur possède son propre profil qui agit comme une sorte de CV, puisqu’on va y retrouver notre travail et nos contributions aux autres projets. Cela peut permettre de savoir si le développeur est actif ou de savoir en quels langages il code, dans l’éventualité de le recruter, par exemple.



Chaque profil GitHub possède un récap de ses commit pour chaque année

Le site offre de nombreuses fonctionnalités habituellement retrouvées sur les réseaux sociaux comme les flux, la possibilité de suivre des personnes ou des projets ainsi que des graphes de réseaux pour les dépôts. GitHub offre aussi la possibilité de créer un wiki et une page web pour chaque dépôt.

On peut ouvrir des issues : ce sont des fils de discussions où les personnes peuvent signaler des bugs, demander des fonctionnalités, ou juste poser des questions.

Le site propose aussi une partie explore, qui permet selon une recherche de voir tous les dépôts, fichiers de code, commit, issues, discussions, etc qui contiennent ce mot-clé, ou simplement de se voir recommander des dépôts selon nos langages de programmation utilisés, etc …

Conclusion

Pour conclure, on peut dire que même s'il existe des alternatives à GitHub telles que GitLab, GitHub se remarque bien plus sur le plan communautaire, qui va offrir beaucoup plus de possibilités de réseautage, et de codage social. GitHub ravira donc ceux qui cherchent la collaboration avec des développeurs extérieurs à son équipe.

Références

https://fr.slideshare.net/ThibaultVlacich/prsentation-git-github?from_action=save

https://www.pierre-giraud.com/git-github-apprendre-cours/presentation-git-github/

https://kinsta.com/fr/base-de-connaissances/base-de-connaissances-github/

https://www.howtogeek.com/180167/htg-explains-what-is-github-and-what-do-geeks-use-it-for/

https://www.youtube.com/watch?v=w3jLJU7DT5E

https://github.com/pricing

https://apiumhub.com/tech-blog-barcelona/using-github/