

GITHUB

Systeme de versioning

VEILLE TECHNOLOGIQUE

Nom : **Moukam Brandon-cury**

Date : **17-11-2024**

Table des matières

I-	INTRODUCTION.....	3
II-	QU'EST-CE QUE GITHUB ?	3
III-	PRINCIPALES FONCTIONNALITÉS	4
1-	Repos (Référentiels).....	4
2-	Branches de fonctionnalités	4
3-	Pull Requests ou Demandes de tirage	4
4-	Issues	4
5-	Actions	6
IV-	AVANTAGES DE GITHUB	6
1.	Collaboration Facilitée	6
2.	Avantage de GitHub en Action.....	6
3.	Historique des Modifications	7
4.	Sécurité.....	7
V-	UTILISATIONS COMMUNES	8
VI-	COMMENT DÉMARRER ?	8
1-	Création d'un compte GitHub et Installation de Git	8
2-	Créer un nouveau référentiel.....	9
2-	Cloner le référentiel.....	9
3-	Faire des commits.....	9
4-	Pousser vers GitHub.....	9
VII-	CONCLUSION	10
	BIBLIOGRAPHIE	10

I- INTRODUCTION

Dans ce cours, nous allons commencer par expliquer ce qu'est Git. Pour cela, nous allons définir ce qu'est un système de gestion de version et comprendre en quoi Git est différent des autres systèmes de gestion de version et pourquoi il est si populaire.

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

Un logiciel de versioning est un logiciel qui permet de conserver un historique des modifications effectuées sur un projet.

Parmi les logiciels de gestion de versions, Git est le leader incontesté et il est donc indispensable pour tout développeur de savoir utiliser Git.

Modèles des logiciels de gestion de version :

- **Le modèle centralisé** : le code source du projet est stocké sur un serveur distant et les modifications y sont directement effectuées ;
- **Le modèle décentralisé** : le code source du projet est également stocké sur un serveur distant et chaque utilisateur est invité à télécharger et à héberger l'intégralité du code source du projet sur sa propre machine.

II- QU'EST-CE QUE GITHUB ?

GitHub est un service en ligne qui permet d'héberger des dépôts ou repo Git. C'est le plus grand hébergeur de dépôts Git du monde.

Pour récapituler, et afin d'être bien clair sur ce point : Git est un logiciel de gestion de version tandis que GitHub est un service en ligne d'hébergement de dépôts Git qui fait office de serveur central pour ces dépôts.

III- PRINCIPALES FONCTIONNALITÉS

1- Repos (Référentiels)

Stockage et gestion des projets.

2- Branches de fonctionnalités

Ils permettent de travailler sur des fonctionnalités séparées sans affecter la branche principale. GitHub offre des branches de fonctionnalités, qui permettent aux développeurs de créer des dépôts locaux isolés pour apporter des modifications au code sans affecter la branche principale.

3- Pull Requests ou Demandes de tirage

Permettent aux développeurs de proposer des modifications à d'autres personnes travaillant sur le même projet. Cela facilite la collaboration et l'examen du code.

4- Issues

Les issues permettent le suivi des bugs et des demandes de fonctionnalités.

Sur GitHub, on distingue plusieurs types d'étiquettes d'issues : **Bug, enchantement, feature request**

a- Les issues étiquetées en Bug :

Les issues bug permettent de signaler et de gérer les bugs de manière centralisée et transparente.

- **Signalement des Bugs :**
 - Lorsque quelqu'un découvre un bug, il peut créer une issue pour le signaler.
 - L'issue doit inclure une description détaillée du problème, y compris des étapes pour le reproduire, des captures d'écran, et tout autre contexte pertinent.
- **Discussion et Confirmation :**
 - D'autres membres de l'équipe peuvent commenter l'issue pour poser des questions, confirmer le problème ou fournir des informations supplémentaires.
 - Le mainteneur du projet peut étiqueter l'issue comme "bug" pour le classer.

- Assignment :

- Le bug peut être assigné à un développeur ou à une équipe spécifique qui sera responsable de le corriger.
- Les assignés peuvent ajouter des commentaires, des commits ou des pull requests liés à l'issue pour suivre les progrès.

- Correction et Fermeture :

- Une fois que le bug est corrigé, le développeur peut lier la pull request de correction à l'issue en utilisant des mots-clés comme "fixes #issue_number".
- Lorsque la pull request est fusionnée, l'issue est automatiquement fermée, indiquant que le bug a été résolu.

b- Demandes de Fonctionnalités

Les issues sont également utilisées pour demander de nouvelles fonctionnalités ou des améliorations.

1. Propositions :

- Un utilisateur ou un développeur peut créer une issue pour proposer une nouvelle fonctionnalité ou une amélioration.
- La description doit inclure une explication de la demande, son utilité, et éventuellement des exemples ou des spécifications.

2. Discussion et Validation :

- La communauté et les mainteneurs peuvent commenter pour discuter de la faisabilité, des avantages, et des inconvénients de la demande.
- Le mainteneur peut étiqueter l'issue comme "enhancement" ou "feature request" pour la classer.

3. Prioritisation :

- Les mainteneurs peuvent prioriser les demandes de fonctionnalités en fonction des ressources disponibles et de l'importance de la demande.
- Les issues peuvent être associées à des milestones pour indiquer quand elles sont prévues pour être traitées.

4. Développement et Fermeture :

- Une fois la demande validée, elle peut être assignée à un développeur ou une équipe pour être implémentée.
- Lorsque la nouvelle fonctionnalité est développée et intégrée via une pull request, l'issue peut être fermée.

5- Actions

Un **workflow**, aussi appelé flux de travaux ou flux opérationnel, désigne une suite de tâches ou d'opérations qui doivent être réalisées par un individu ou un groupe d'individus selon un ordre spécifique.

GitHub Actions permet de créer des workflows personnalisés pour automatiser des tâches telles que l'intégration continue (CI), le déploiement continu (CD)

IV- AVANTAGES DE GITHUB

1. Collaboration Facilitée

- **Contrôle de Version** : GitHub utilise Git, un système de contrôle de version distribué, permettant à plusieurs développeurs de travailler sur le même projet simultanément sans conflits.
- **Pull Requests** : Facilite les révisions de code, permettant aux équipes de discuter, d'examiner et de fusionner des modifications de code de manière structurée.

2. Avantage de GitHub en Action

Déploiement Continu (CD) :

- Déployer automatiquement votre application sur un serveur ou un service cloud à chaque mise à jour validée.

Intégration Continue (CI) :

- Construire et tester votre code à chaque push ou pull request pour s'assurer qu'il n'y a pas de régressions.

Automatisation des Tâches :

- Planifier des tâches comme le nettoyage des branches obsolètes, la génération de documentation, ou l'envoi de notifications.

3. Historique des Modifications

Sur GitHub, les éléments qui permettent d'effectuer le suivi complet des changements sont :

Commits :

Chaque changement apporté à un fichier est enregistré sous forme de commit. Un commit inclut un message descriptif, l'auteur des modifications, et un identifiant unique appelé SHA.

Branches :

- Les commits sont organisés dans des branches. Une branche est une séquence de commits qui permet de travailler sur des fonctionnalités ou des correctifs de manière isolée.
- La branche principale est souvent appelée main ou master.

Pull Requests :

- Un pull request est une demande de fusion des modifications d'une branche dans une autre. Elle permet de réviser et discuter des changements avant de les intégrer.

Diffs :

- Les diffs (différences) montrent les modifications apportées aux fichiers entre deux commits ou deux branches. Ils permettent de voir exactement ce qui a été ajouté, modifié ou supprimé.

Blame :

- La fonctionnalité "blame" permet de voir l'historique des modifications ligne par ligne pour un fichier donné. Elle montre qui a modifié chaque ligne et dans quel commit.
- Cela aide à identifier l'auteur des modifications et à comprendre le contexte des changements.

4. Sécurité

Pour garantir la sécurité des projets, GitHub utilise les : **Permissions, rôles et Invitations**

a- Les permissions :

- GitHub permet de définir des permissions spécifiques pour chaque utilisateur ou groupe. Par exemple, vous pouvez accorder des permissions de lecture, d'écriture, ou d'administration.
- Les permissions peuvent être définies au niveau du dépôt, de la branche, ou même de fichiers spécifiques.

b- Les rôles :

- GitHub propose des rôles prédéfinis comme "Maintainer", "Contributor", et "Viewer", mais vous pouvez également créer des rôles personnalisés pour répondre à vos besoins spécifiques.
- Les rôles déterminent les permissions que les utilisateurs ont sur le dépôt.

c- Les invitations :

- Vous pouvez inviter des utilisateurs à rejoindre votre dépôt en leur envoyant un lien d'invitation par e-mail.
- Les invités doivent accepter l'invitation avant de pouvoir accéder au dépôt.

V- UTILISATIONS COMMUNES

• Gestion de Projet

L'Organisation des tâches et le suivi des progrès s'effectuent grâce aux Issues, Projets, Pull Requests,

• Développement Open Source

Le développement en open source permet de donner la possibilité à tout le monde de contribuer au projet.

• Portfolio de Code

GitHub peut être utilisé pour présenter ses projets et compétences au client et à son employeur, ce qui donne une vision globale des projets réalisés et du degré de maîtrise.

VI- COMMENT DÉMARRER ?

1- Création d'un compte GitHub et Installation de Git

- La création de compte GitHub est gratuite. L'adresse web pour créer un compte est :

https://github.com/signup?ref_cta=Sign+up&ref_loc=header+logged+out&ref_page=%2F&source=header-home

- Télécharger et installer la dernière version sur le site officiel <https://git-scm.com/downloads>

2- Créer un nouveau référentiel

Les étapes pour créer un nouveau référentiel sont :

- **Connectez-vous** à votre compte GitHub sur github.com ;
- **Naviguez** à la page d'accueil de votre profil ;
- **Cliquez** sur le bouton vert New ou allez directement à <https://github.com/new>
- **Nom du Référentiel** : Donnez un nom à votre nouveau projet dans le champ Repository name ;
- **Description (Optionnelle)** : Vous pouvez ajouter une description pour donner un contexte sur le projet ;
- **Visibilité** : Choisissez entre Public (visible par tout le monde) ou Private (visible uniquement par vous et les collaborateurs que vous invitez)
- **Initialiser ce référentiel avec** : Cochez la case Add a README file. Le fichier README est important car il sert de présentation du projet.
- **Cliquez** sur le bouton Create repository pour créer le référentiel.

2- Cloner le référentiel

Pour cloner un référentiel GitHub, il faut :

- **Copie l'URL** : Sous l'onglet "HTTPS", copie l'URL du repository
- Dans notre terminal, il faut exécuté la commande le clonage : `git clone [URL du repository]`
- **Vérifie le Clonage** : Une fois le clonage terminé, un nouveau dossier sera créé avec le même nom que le repository.
- Navigue dans ce dossier en exécutant la commande : `cd nom-repository`

3- Faire des commits

Après des modifications sur les fichiers, nous pouvons les comités en effectuant les commandes suivantes dans notre terminale :

- `git add .` // Ajoute tous les fichiers modifiés dans le répertoire courant au staging area.
- `git commit -m "Description de tes changements"` // Crée un commit avec les fichiers ajoutés au staging area et inclut un message décrivant les changements.

4- Pousser vers GitHub

Pour enfin enregistrer les modifications sur GitHub sur la branche principale (main), il suffit d'exécuter la commande :

- `git push origin main`

VII- CONCLUSION

En résumé, GitHub est bien plus qu'une simple plateforme de gestion de code ; c'est un outil puissant qui facilite la collaboration, l'organisation et la gestion de projets de développement logiciel. Grâce à ses fonctionnalités robustes comme le contrôle de version, les issues, les pull requests, et GitHub Actions, GitHub offre un environnement complet pour développer, tester, déployer et maintenir des projets de manière efficace et sécurisée.

BIBLIOGRAPHIE

<https://docs.github.com/>

<https://docs.github.com/fr/get-started/start-your-journey/hello-world>

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

<https://dessein-tech.com/t/quest-ce-que-github-et-comment-fonctionne-t-il/671>