

Compte rendu d'activité

TP : Utilisation de Github et aide utilisateur

03/2022

Contexte

Travaillant dans une entreprise de services numériques (ESN) qui accompagne ses clients dans la mise en œuvre de solutions numériques et mobiles depuis leur conception jusqu'à leur intégration dans le système d'information du client, il est nécessaire de mettre en place le déploiement d'une application cliente, à sa validation technique et à sa mise en production. Vous assistez les utilisateurs dans la prise en main des nouvelles solutions logicielles mises en place à partir du serveur créé sous machine virtuelle.

Expressions des besoins

- Il est souhaité que les mises à jour doivent se faire automatiquement avec Git.
- La nouvelle version permet de contrôler qu'un email saisie dans la partie abonnement ne comporte pas d'erreur dans son format. Il faut écrire les scénarios de tests permettant de tester cette nouvelle fonctionnalité.
- Mettre à jour le code actuellement sur le serveur à partir de la dernière version présente dans la « branche » de développement sur GitHub nommée « dev ». Pour ce faire, il faut donc remplacer les fichiers actuels, par ceux présents dans la dernière version de la branche « dev ».
- Après avoir Fork le projet, tester la nouvelle fonctionnalité de la branche dev de recherche accessible sur toutes les pages. Créer des issues depuis la plateforme github en cas d'anomalie détecté.
- Enfin, une vidéo de présentation de la nouvelle fonctionnalité sera mise en place avec le logiciel OBS pour une documentation utilisateur. Visible à ce [lien](#)

Informations techniques

Le code source de l'application est accessible à ce [lien](#).

Realisation

Récupération du code source avec git

Le code source de l'application étant accessible à ce [lien](#), un dossier de dépôt a été créé en local. Depuis l'invite de commande de git bash, ce dossier a été converti en dépôt par la commande : **git init**

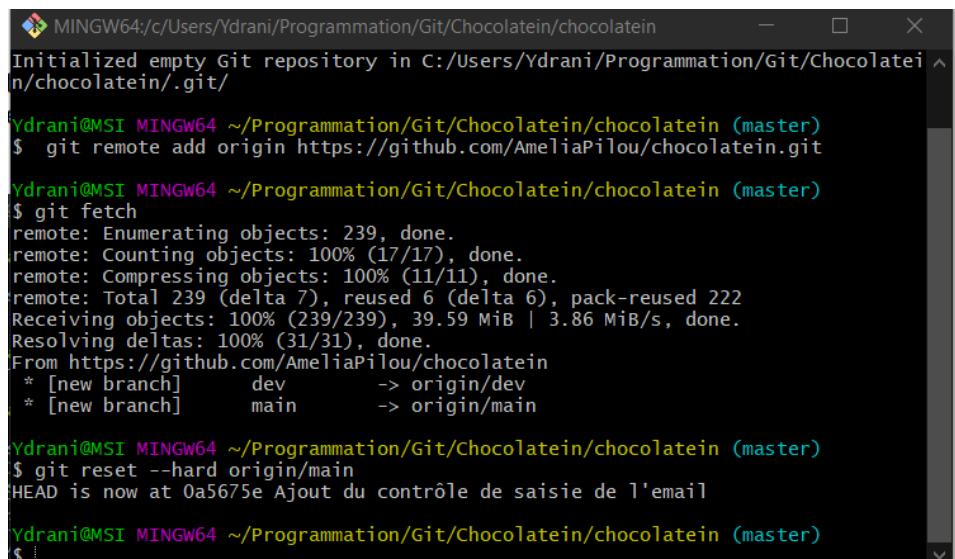
Ensuite, le code source a été récupéré par la commande de connexion **remote add**.

Puis, un téléchargement de la dernière version a été effectué avec : **git fetch**

Enfin, les fichiers de cette dernière version ont remplacé les anciens avec la commande :

git reset --hard origin/main

où **git reset** permet de forcer la mise à jour du dépôt local à partir de n'importe quelle branche d'un dépôt.



```
MINGW64: c:/Users/Ydrani/Programmation/Git/Chocolatein/chocolatein
Initialized empty Git repository in C:/Users/Ydrani/Programmation/Git/Chocolatein/chocolatein/.git/

Ydrani@MSI MINGW64 ~/Programmation/Git/Chocolatein/chocolatein (master)
$ git remote add origin https://github.com/AmeliaPilou/chocolatein.git

Ydrani@MSI MINGW64 ~/Programmation/Git/Chocolatein/chocolatein (master)
$ git fetch
remote: Enumerating objects: 239, done.
remote: Counting objects: 100% (17/17), done.
remote: Compressing objects: 100% (11/11), done.
remote: Total 239 (delta 7), reused 6 (delta 6), pack-reused 222
Receiving objects: 100% (239/239), 39.59 MiB | 3.86 MiB/s, done.
Resolving deltas: 100% (31/31), done.
From https://github.com/AmeliaPilou/chocolatein
 * [new branch]      dev       -> origin/dev
 * [new branch]      main      -> origin/main

Ydrani@MSI MINGW64 ~/Programmation/Git/Chocolatein/chocolatein (master)
$ git reset --hard origin/main
HEAD is now at 0a5675e Ajout du contrôle de saisie de l'email

Ydrani@MSI MINGW64 ~/Programmation/Git/Chocolatein/chocolatein (master)
$
```

Figure 1: Console git bash avec les résultats des successions de commandes afin de récupérer la dernière version du dépôt

Aussi, il est possible de lire le dernier message de commit : Ajout du contrôle de saisie de l'email.

Scénario test des emails

Un email est normalement de la forme suivante : **chaîneDeCaractères@domaine.aa**

Selon la norme RFC 822, tous les caractères non valides sont :

- les lettres accentuées
- les caractères de contrôle (CTRL + touche clavier)
- les espaces
- les signes spéciaux comme (<>@,;:"'[]|ç%& ,
- Pas de domaine de longueur inférieur à 2

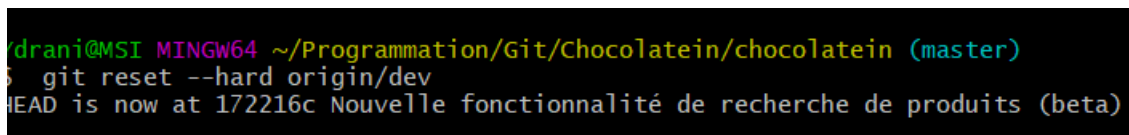
Ainsi, un ensemble de test pourrait être les suivants :

```
assertEquals(true,formulaire.validerEmail("dupont.marie@exemple.fr");
assertEquals(true,formulaire.validerEmail("n-23mE_.l0Ast_9Na-Me@valide01-1234567.frrrrrrrrrrrrrrrrrr"));

assertEquals(false,formulaire.validerEmail("@exemple.fr") ;
assertEquals(false, formulaire.validerEmail("exemple.fr") ;
assertEquals(false, formulaire.validerEmail("n-234Me@inValide[<01-.f") ;
assertEquals(false, formulaire.validerEmail("n-23\\E_.\ "true\ "n-E()._.\ « 0Ast_9-Na4M@inval01-124579.com") ;
```

Mettre à jour le code

Mettre à jour le code actuellement sur le serveur à partir de la dernière version présente dans la « branche » de développement sur GitHub nommée « dev ». Pour ce faire, il faut donc remplacer les fichiers actuels, par ceux présents dans la dernière version de la branche « dev », en utilisant le code suivant : **git reset --hard origin/dev**



```
drani@MSI MINGW64 ~/Programmation/Git/Chocolatein/chocolatein (master)
$ git reset --hard origin/dev
HEAD is now at 172216c Nouvelle fonctionnalité de recherche de produits (beta)
```

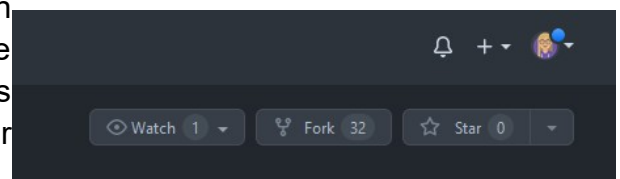
Figure 2: Réponse de la console git bash apres insertion de la commande

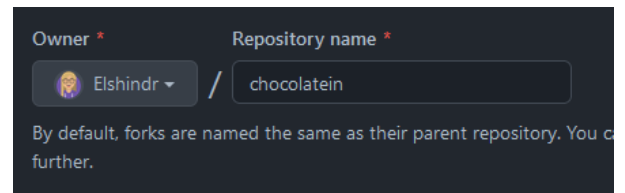
De même, le dernier message de commit de la branche s'affiche : Nouvelle fonctionnalité de recherche de produit.

Détection d'anomalies

Récupération et installation

Afin de fork le projet, il est nécessaire d'avoir un compte Github et de se rendre sur la page du projet en question, [ici](#). C'est bien la branche dev qu'il convient de récupérer. Sur la page en question, il faut cliquer sur le bouton de fork en haut à droite. A la suite, une fenêtre de création s'ouvre afin de créer un dépôt depuis son propre compte lié à ce projet, après avoir saisi un nom, il faut cliquer sur create Fork.

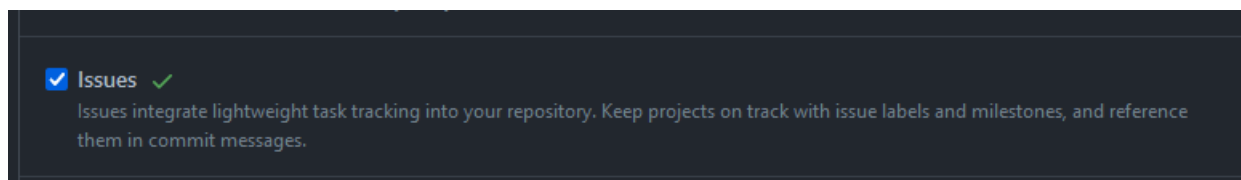




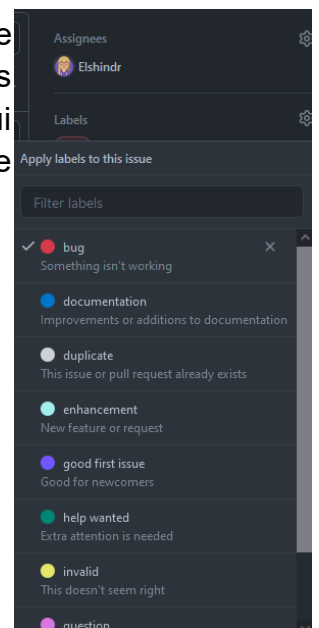
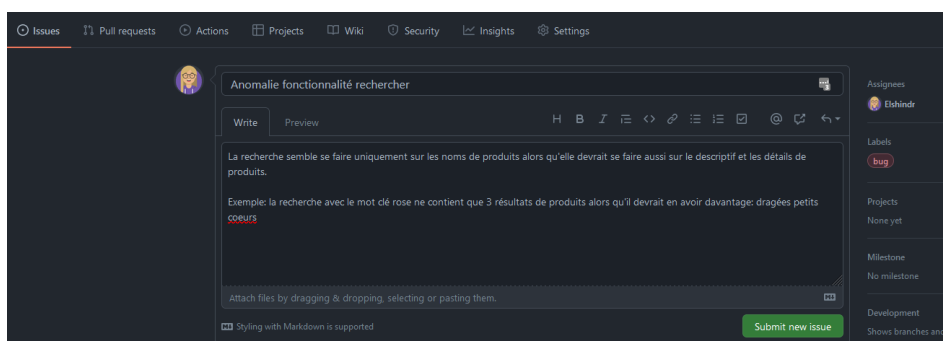
Après avoir Fork le projet, il est possible de récupérer le clone du nouveau dépôt du projet, puis d'installer ce dépôt soit sur un serveur virtuel sous VM, soit via WampServer. Ici, le projet a été glissé-déposé dans la VM de serveur Apache préalablement créé via le logiciel FilaZilla Client dans son dossier **/var/www/html**

Tests de la nouvelle fonctionnalité

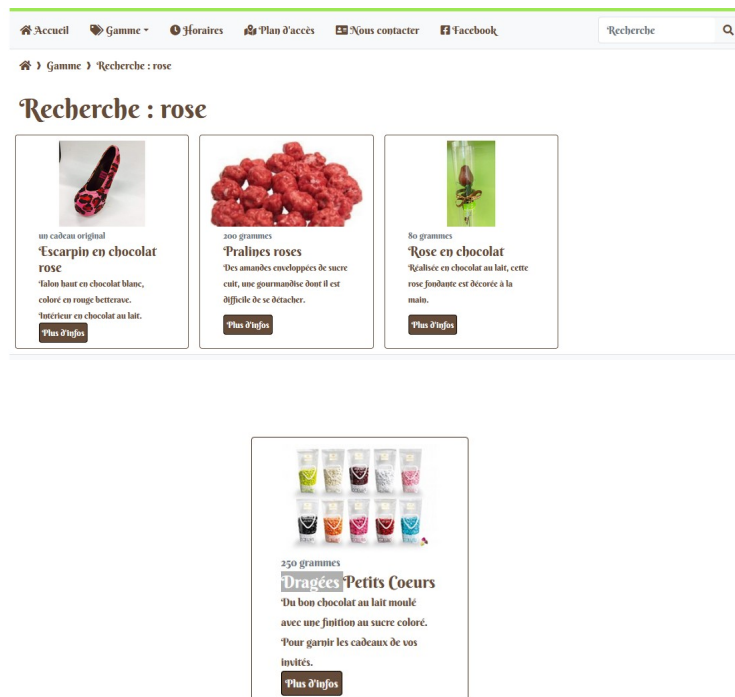
Il est possible d'activer la fonction issue depuis son compte GitHub, dans Settings, Features puis en cochant la case Issues.



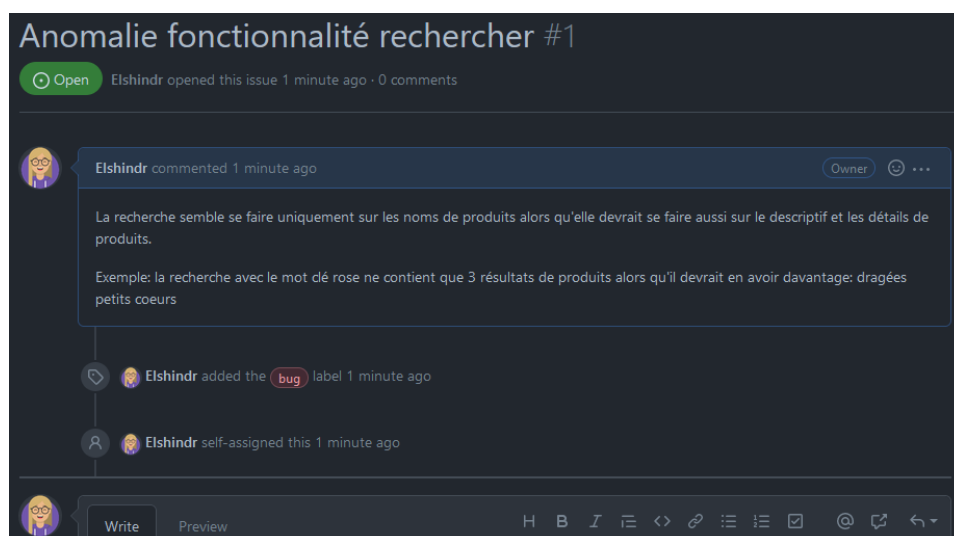
En se rendant dans l'onglet Issue du dépôt distant, un bouton 'New issue' permet de créer une nouvelle Issue en cas de bug détecté. Il est alors possible de reporter ce bug en le décrivant, d'ailleurs il est possible de lui donner un label qui est une étiquette avec une désignation précise permettant d'identifier rapidement le type d'issue, ici il est mis sur bug.



En utilisant la fonction de recherche, il a été possible de constater que la recherche ne se fait que sur les noms de produits et non pas sur sa description ou ses détails. Ce dysfonctionnement sera ainsi reporté. Par exemple, le mot rose ne contient que 3 résultats alors qu'il devrait en avoir plus, comme les « dragées petits coeurs »



Ainsi, après avoir reporté ce bug sur la plateforme Github, il est possible de consulter cette issue à ce [lien](#).



Liste des compétences couvertes

Mettre à disposition des utilisateurs un service informatique :

- Réaliser les tests d'intégration et d'acceptation d'un service
- Accompagner les utilisateurs dans la mise en place d'un service