

Gestion de projets sur GitHub

5 JUILLET

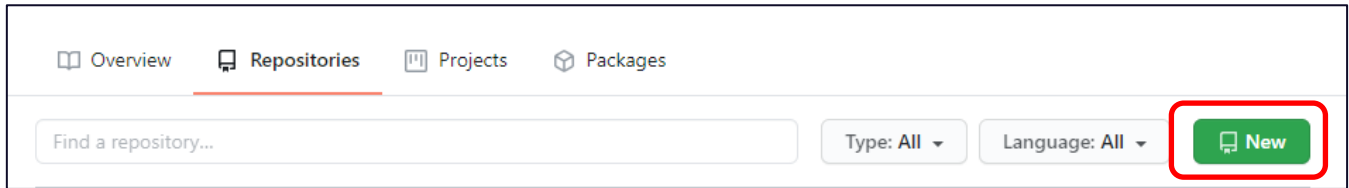
Concepteur Développeur d'Application

Créé par : Jonathan JEANNIARD

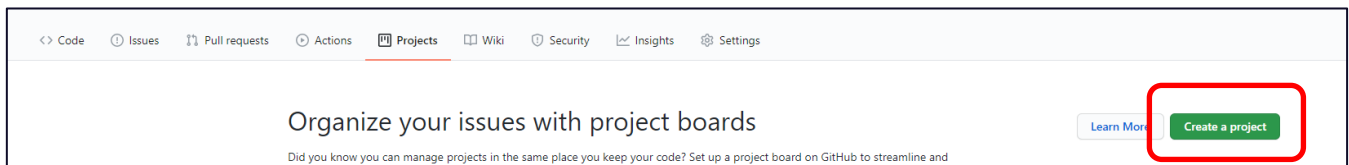
Contributeurs : Aurélien BOUDIER



1. Création d'un repository sur GitHub

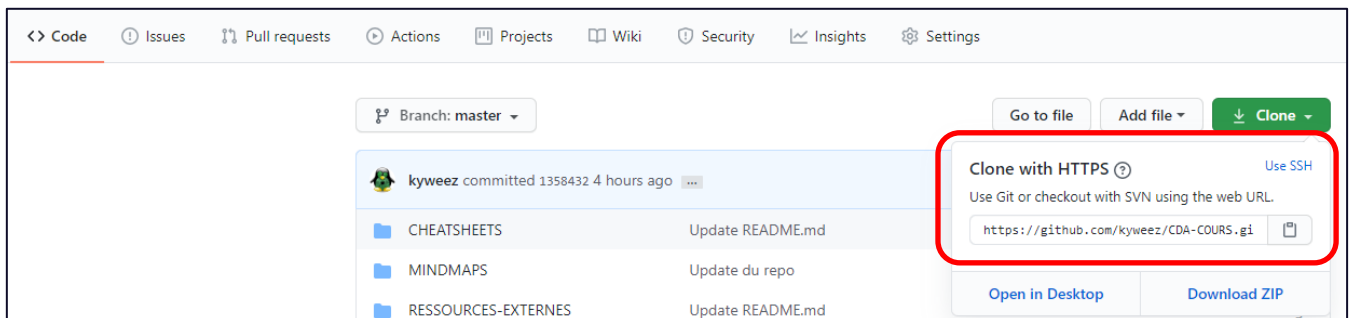


2. Création d'un projet



3. Récupération du repository distant en local

Dans un premier temps, il faut s'assurer que tous les collaborateurs du projet ont initialisé le dépôt distant en local (git clone).



4. Exemples de branches

Les branches de référence sont les suivantes :

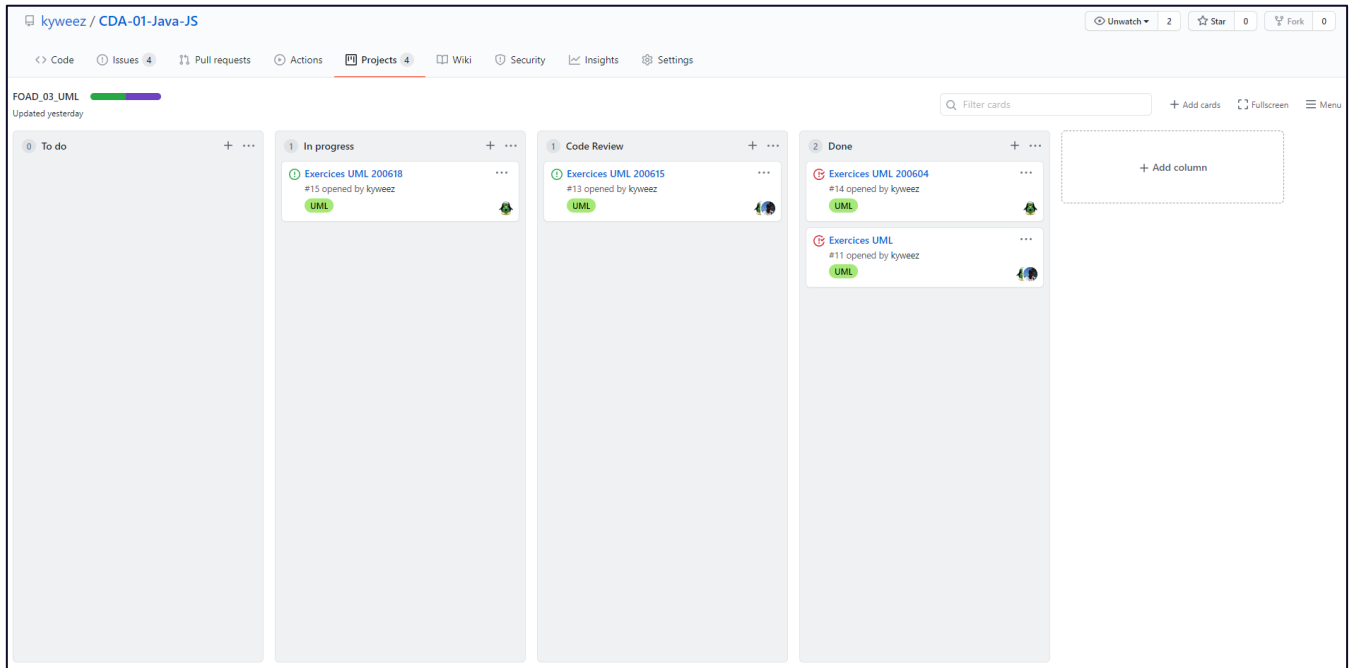
- Master : contient le rendu final et fonctionnel du projet
- Develop : point de rassemblement de tous les merges des différentes features passés en review
- Features : branche de développement des features. Ces branches sont push sur la branche develop après un code review de la team.

5. Le Kanban du projet

Les projets permettent une disposition de différentes colonnes en mode Kanban. Dans nos projets nous utiliserons 4 colonnes :

- To Do
- In progress
- Code Review
- Done

Ces 4 colonnes permettent d'avoir un suivi en temps réel du projet.



a. To Do

La colonne "Todo" est paramétrée de telle sorte à récupérer les tickets de l'onglet "issues" et les transformer en tâche à faire.

b. In Progress

Chaque ticket est attribué à un (ou plusieurs) développeur(s), qui glissera ce ticket dans la colonne "In progress" pour montrer qu'il travaille dessus.

c. Code review

Après avoir développé la feature et rempli le ticket dans son ensemble, le développeur glisse le ticket dans "Code Review" et l'associe à un Pull Request afin d'en discuter avec les collègues du projet.

d. Done

Une fois la code review effectuée, le merge est fait avec la branche supérieure et le ticket est fermé puis glissé dans la colonne "Done".

6. Le travail sur des branches

Pour faciliter le travail en équipe, il est déconseillé de travailler tous ensemble sur la même branche. Il est d'ailleurs fréquent qu'on ne puisse pas travailler sur la branche master ou develop car ces dernières sont protégées pour des raisons évidentes de sécurité. La branche master contiendra une version finale fonctionnelle du projet, qui sera en production. La branche develop contiendra les nouvelles features. Cette branche sera souvent fonctionnelle, mais pas encore pushed sur la branche master car elle nécessitera plus de tests ou l'ajout supplémentaires de features avant de faire évoluer la version définitive.

C'est pour cette raison qu'il est recommandé de créer une branche par nouvelle feature, que nous mergerons avec la branche develop quand cette feature sera fonctionnelle et que les effets de bords seront contrôlés. Il faudra souvent faire un pull request vers la branche develop.

Pour la gestion de branches, git flow est un excellent outil. [Un tutoriel est disponible ici](#).

7. Le pull request

Lorsque l'on a fini notre travail sur notre branche, on souhaite faire une pull request vers la branche develop. Rien de plus simple : 2 façons de faire :

The screenshot shows a GitHub repository page for 'feature/gestion-de-projet_adding-stuff_kyweez'. A red box highlights the 'Compare & pull request' button, with an annotation 'Cliquer ici' (Click here) pointing to it. A green box highlights the 'Pull request' button, with an annotation 'Se positionner sur notre branche et cliquer sur pull' (Position on our branch and click on pull) pointing to it. The repository details show 60 commits, 9 branches, and 0 tags. The README.md file is visible, titled 'CDA-COURS', with a description: 'Ce dépôt a pour objectif de regrouper les ressources que chacun pense être utiles, propres et cohérentes avec les cours que nous suivons en CDA.' The right sidebar shows sections for Releases, Packages, and Contributors.

Ensuite s'assurer de faire la pull request de la bonne branche (1) vers la bonne branche (2) ! Marquer un petit commentaire pour les collègues et créer la pull request (3) :

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

The screenshot shows the 'Open a pull request' form. A red box with the number '2' highlights the 'base: develop' dropdown menu. A red box with the number '1' highlights the 'compare: feature/gestion-de-projet_addi...' dropdown menu. A red box with the number '3' highlights the 'Create pull request' button. The form includes a title field with the text '#2-gestion-de-projet / Init modif', a 'Write' tab, a 'Preview' tab, and a 'Leave a comment' text area. The bottom of the form has a note: 'Attach files by dragging & dropping, selecting or pasting them.'

8. Le code review

Une fois la pull request faite si le repository est paramétré correctement, des reviewers seront nécessaires pour valider la pull request avant de faire un merge.

- On peut assigner des reviewers. (1)
- On peut relier la pull request à un projet (dans un kanban de projet automatique, la tâche se déplacera toute seule). (2)
- On peut relier la pull request à un ticket (qui se clôturera si elle est acceptée). (3)
- Si les reviewers nécessaires (2 dans ce cas-ci) ont accepté les modifications, on sera en mesure de cliquer sur merge. (4)

#2-gestion-de-projet / Init modif #21

kyweez wants to merge 1 commit into develop from feature/gestion-de-projet_adding-stuff_kyweez

Conversation 0 Commits 1 Checks 0 Files changed 3 +0 -0

kyweez commented 1 minute ago
No description provided.

kyweez commented now
Pull request faite pour le cours. Elle sera supprimée après... C'est pour illustrer.

Add more commits by pushing to the feature/gestion-de-projet_adding-stuff_kyweez branch on kyweez/CDA-RESSOURCES.

Review required
At least 2 approving reviews are required by reviewers with write access. [Learn more.](#)

Merging is blocked
Merging can be performed automatically with 2 approving reviews.

As an administrator, you may still merge this pull request.

1 Reviewers
No reviews—at least 2 approving reviews are required.
Still in progress? Convert to draft

2 Projects
None yet

3 Linked issues
Successfully merging this pull request may close these issues.
None yet

4 Merge pull request

Côté reviewer :

Un reviewer se doit de vérifier les changements apportés. Il peut cliquer sur les onglets « Commits » (1) ou « Files changed » (2) pour observer les modifications. Une fois les vérifications faites, cliquer sur « Review changes » (3) :

Update README.md #13

kyweez wants to merge 1 commit into master from patch-1

Conversation 0 Commits 1 Checks 0 Files changed 1 +1 -1

Changes from all commits File filter... Jump to... 0 / 1 files viewed

1 Commits

2 Files changed

3 Review changes

2 README.md

@@ -1,4 +1,4 @@

1 1 # DMM_2005

2 2

3 3 Dépôt DMM et CDA 2005.

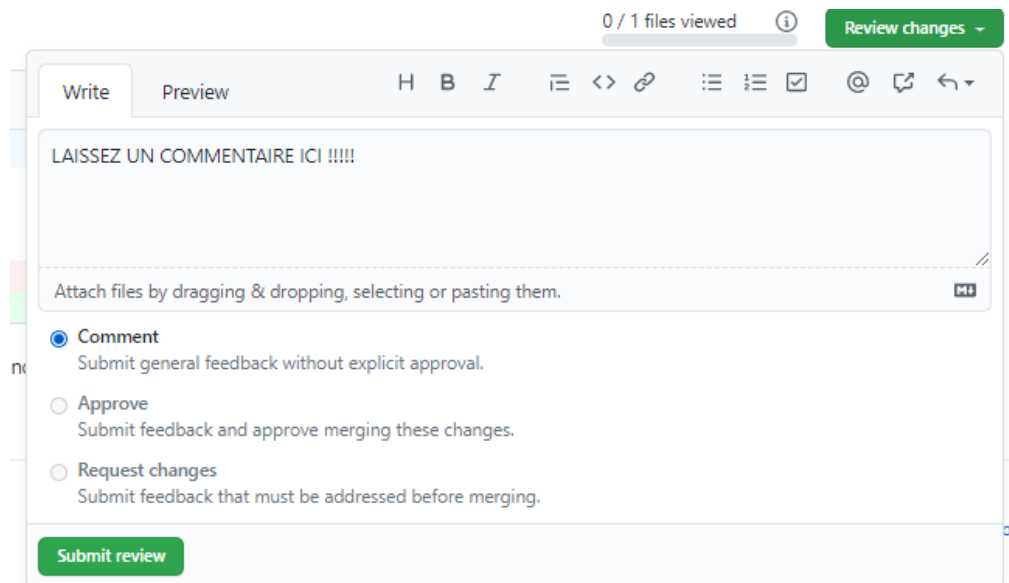
4 4 - Les codes démos et corrections y seront centralisés.

4 4 + Les codes démos et corrections y seront centralisés

Changements apportés

ProTip! Use [n] and [p] to navigate between commits in a pull request.

On peut simplement commenter, ou bien on peut approuver, ou encore demander des changements. Quoi qu'il en soit, il faut laisser un commentaire pour valider la review, quel que soit le statut qu'on laisse :



The screenshot shows a code review interface. At the top, there's a status bar with "0 / 1 files viewed" and a "Review changes" button. Below this is a toolbar with "Write" and "Preview" tabs, and various formatting icons (H, B, I, list, link, etc.). The main area is a large text box containing the placeholder text "LAISSEZ UN COMMENTAIRE ICI !!!!!". Below the text box is a dashed line and the instruction "Attach files by dragging & dropping, selecting or pasting them." with a small icon. At the bottom, there are three radio button options: "Comment" (selected), "Approve", and "Request changes". Each option has a brief description: "Submit general feedback without explicit approval.", "Submit feedback and approve merging these changes.", and "Submit feedback that must be addressed before merging." respectively. A green "Submit review" button is at the very bottom.

0 / 1 files viewed Review changes

Write Preview H B I ≡ <> 🔗 ≡ ≡ ☑ @ 🔗 ↩

LAISSEZ UN COMMENTAIRE ICI !!!!!

Attach files by dragging & dropping, selecting or pasting them. 📎

☒ **Comment**
Submit general feedback without explicit approval.

☐ **Approve**
Submit feedback and approve merging these changes.

☐ **Request changes**
Submit feedback that must be addressed before merging.

Submit review

Crédits :

Action	Auteur	Descriptif	Date
Création du document	Jonathan JEANNIARD	Gestion de projet sur GitHub	22/06/20
Correction	Aurélien BOUDIER	Correction de l'orthographe et remise en page selon la trame	23/06/20
Ajout	Aurélien BOUDIER	Ajout de la partie sur le code review	05/07/20