GitLab Classroom

Depuis environ deux ans, Télécom Nancy a déployé au sein de son infrastructure une instance de GitLab qui remplace la forge précédemment utilisée par les étudiants. C'est un outil central qui permet aux développeurs de partager leurs codes et de collaborer à plusieurs sur un même projet. En somme, GitLab héberge de dépôts git à l'instar de la plateforme GitHub.

Les étudiants de Télécom Nancy sont amenés à utiliser très régulièrement GitLab que ce soit pour les travaux pratiques ou les projets. Le nombre et la qualité des commits poussés par les étudiants sont des indicateurs permettant d'apprécier l'implication de ces derniers dans les travaux/projets proposés. De cette manière, l'enseignant peut plus rapidement repérer les étudiants en difficulté.

Malheureusement, GitLab n'est pas entièrement adapté aux usages de l'enseignant, la création et la gestion des dépôts se révèlent être fastidieuses surtout lorsque le travail proposé (TP ou projet) est commun à toute une promotion (génération d'un grand nombre de dépôts ...).

Les fonctionnalités minimales attendues

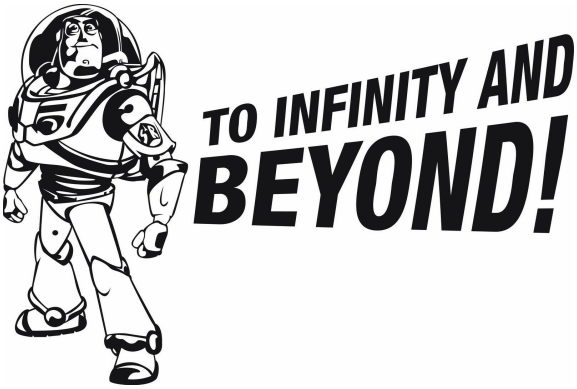
Ce projet vise à concevoir et développer une application web permettant à l'enseignant de gérer les dépôts des étudiants tout au long des activités pédagogiques à l'instar de GitHub Classroom. La nécessité de disposer d'une telle application est un réel besoin qui est décrit par l'issue #26681 de GitLab community.

L'enseignant devra être en mesure :

- de renseigner une clé d'API nécessaire pour que l'application web puisse interagir avec l'instance GitLab
- de créer un lien destiné aux étudiants pour qu'ils puissent s'inscrire à un groupe. Ce groupe pourra être associé à des devoirs.
- de créer ou modifier un devoir (TP / projet … vous pouvez redéfinir à votre guise la terminologie)
    - un devoir est un dépôt GitLab qui peut contenir des supports : code source, fichier pdf, README …
        - Ces supports peuvent être modifiés au cours du temps
- d'attribuer un devoir à un groupe d'étudiants en précisant si ce devoir est individuel ou en groupe (en précisant la cardinalité : binôme / trinôme ...) soit de manière automatique soit manuellement
- d'associer des enseignants à ce devoir via leur identifiant GitLab
- de créer et peupler les dépôts associés aux devoirs après attribution des devoirs aux étudiants
- de générer des statistiques pour chaque dépôt sur la participation des étudiants:
    - Nombre de commits / élève
    - Nombre d'insertions - suppressions / élève

    - *Vous pouvez vous inspirer du projet Gitinspector pour compléter ces statistiques*
- d'organiser de manière cohérente les dépôts, les devoirs, les groupes d'étudiants de façon à :
    - réutiliser le contenu des devoirs d'une année à l'autre
    - accéder rapidement aux dépôts des activités pédagogiques de l'année en cours
    - cette organisation devra être répercutée sur GitLab (vous pouvez par exemple utiliser les projets)

Notez qu'il n'est pas suffisant d'implémenter correctement les fonctionnalités décrites ci-dessus et d'écrire une bonne documentation pour obtenir l'intégralité des points.



Vous devrez proposer des fonctionnalités **innovantes** et ayant un réel intérêt pour l'utilisateur final. L'implémentation d'une telle application soulève des questions sur les besoins en termes d'outils pédagogiques, notamment sur les moyens nécessaires pour accompagner et suivre au mieux les étudiants durant les activités (projets / TP …).
Interrogez-vous sur ces aspects et proposez des solutions **réalisables dans le temps imparti** permettant d'y répondre.

Des fonctionnalités plus avancées **(propositions)**

Pour celles et ceux qui seraient en manque d'inspiration, voici quelques propositions d'idées. Ce ne sont que des propositions, vous êtes libres de modifier l'idée originale ou tout simplement d'en choisir d'autres.

Corrections automatiques avec GitLab CI/CD

Un module d'intégration et de déploiement continu (CI/CD) est disponible sur l'instance GitLab de Télécom Nancy. Ce module permet la mise en place de pipelines de tests qui seront exécutés à chaque modification poussée sur l'instance (git push). La définition et l'organisation des tests sont décrites dans fichier `.gitlab-ci.yml` situé à la racine du dépôt.
Pour mieux comprendre la façon dont le module fonction vous pouvez consultez ces ressources :

Ressources personnelles :

- Introduction aux pratiques DevOps (présentation GLA 2017)
- Présentation des activités (GLA 2017) seules les activités 1 & 2 devraient vous être utiles

- Activité CI Latex

Documentation :

- Configuration of your jobs with .gitlab-ci.yml
- GitLab CI/CD Examples

La correction automatique des TPs est d'ores et déjà possible et même réalisée pour le module de C. Encore une fois, ce qui fait principalement défaut, c'est le manque d'une interface permettant de visualiser globalement les résultats des tests pour tous les dépôts.

Ressources techniques :

- Documentation de l'API REST GitLab : Pipelines API
- Documentation de la librairie "wrapper" python : Pipelines and Jobs

Attention, rappelez-vous que le but du projet n'est pas de maîtriser à la perfection GitLab CI/CD, vous pouvez donc vous aider d'exemples très simples comme : hello-worl-ci, advanced-hello-world-ci pour réaliser cette fonctionnalité.

Interactions via le système d'issues de GitLab

Il est parfois nécessaire de faire des remarques individuelles sur un dépôt, par exemple : le travail n'est pas suffisamment avancé, des fichiers compilés ont été poussés … (Ca vous rappelle quelque chose ? 😈) Le système d'issues se révèle être intéressant à plusieurs égards pour ce cas d'usage. L'étudiant peut répondre à l'issue et fermer l'issue une fois que les problèmes soulevés ont été résolus. D'autre part, l'enseignant a aussi la possibilité d'associer un label à l'issue est donc de la catégoriser. En somme cela permet une interaction riche et individualisée en plus d'habituer l'élève au système d'issues auquel il sera très certainement confronté dans sa carrière future.
A partir de ce constat, il serait intéressant de proposer une interface permettant de faciliter la création, modification et le suivi des issues.

Système de notifications avec l'instance Mattermost de TN

Une instance Mattermost est également disponible au sein de l'école. Une idée intéressante pourrait être de notifier les étudiants via Mattermost par exemple pour prévenir d'un changement de date de rendu ou d'un changement de consigne. (N'hésitez pas à compléter cette idée par d'autres cas d'usage…)

Pourquoi pas une interface pour les étudiants aussi ?

Les élèves peuvent aussi avoir besoin d'une interface offrant par exemple une vue globale des activités pédagogiques dans lesquels ils sont impliqués.

Discutez entre vous et partagez vos idées et vos réflexions… :-)

Consignes et évaluation du projet

Consignes

Ce projet devra être réalisé par groupe de 3/4 personnes. Vous devrez créer un dépôt **privé** avec le préfix `pweb-2k19` sur la **plateforme GitLab de Télécom Nancy** en m'y ajoutant en tant qu'administrateur. Une fois ces deux conditions réunies, l'un des membres de l'équipe devra inscrire le groupe ici.

Ce projet consiste donc à développer une application web qui comprend :

- Un partie client : JS / TS (et tous les frameworks : AngularJS, react …)
- Une partie serveur : La technologie de votre choix

Vous êtes certes libres de choisir les technologies web que vous affectionnez le plus. Mais attention, choisissez des technologies que vous maîtrisez ou à défaut celles vues durant les MOOCs. Vous n'aurez pas le temps de vous familiariser avec une nouvelle technologie et de réaliser le projet dans les délais impartis.

Par ailleurs, même si les technologies choisies diffèrent de celles présentées dans le MOOC, vous devrez tout de même mettre en oeuvre les pratiques et les concepts qui ont été vus tout au long de ce module : persistance en BD, Sécurité, APIs …

Le rendu de ce projet comprend :

- ➔ Un dépôt git dans lequel doit se trouver :
  - ◆ l'intégralité des sources de votre application et les instructions pour la déployer
  - ◆ une documentation technique détaillée sur l'architecture et les choix techniques réalisés
- ➔ Les cas d'usage implémentés présentés au travers d'une **vidéo de 10 minutes**. Pour la capture de la vidéo, vous pouvez utiliser https://screencast-o-matic.com/. La vidéo pourra être partagée sur la plateforme de votre choix, le plus simple étant YouTube ou Dailymotion. Le lien devra apparaître très **clairement** dans le fichier `README` du dépôt.
- ➔ Un lien vers votre application déployée dans le fichier `README`.

Le travail collectif nécessite de s'organiser. Votre dépôt git doit refléter cette organisation. Par organisation, j'entends : découpage du projet en blocs de fonctionnalités cohérentes, utilisation des branches pour le développement de chaque fonctionnalité (Git Branching - Branching Workflows) … Lorsqu'un ensemble cohérent de fonctionnalités a été implémenté, vous devez publier une `RELEASE` qui comprend : la liste de fonctionnalités implémentées, le code source et les instructions pour l'exécuter. Vous devrez faire au minimum 3 `RELEASES`. L'application publiée par une `RELEASE` doit être fonctionnelle.

Le projet est à rendre dans son intégralité pour le vendredi 10 mai.

Ressources

GitLab propose une API REST cependant, il est fortement conseillé d'utiliser une librairie "wrapper" qui abstrait son utilisation : python-gitlab pour python par exemple.

Aidez-vous les uns les autres, mais ne trichez pas

Aider et se faire aider est encouragé, mais l'honnêteté académique vous impose de lister vos aides. Tout manquement à cette règle est une tricherie et sera sanctionné comme il se doit. Par se faire aider, on entend : avoir une discussion sur le design du code, y compris sur des détails techniques et/ou les structures de données à mettre en œuvre. Par tricher, on entend : copier ou recopier du code ou encore lire le code de quelqu'un d'autre pour s'en inspirer. Nous testerons l'originalité de votre travail par rapport aux autres projets rendus.[1]

Evaluation

Une grande partie de l'évaluation portera sur deux aspects majeurs: la mise en oeuvre des concepts techniques abordés dans le module et le développement d'une application offrant des fonctionnalités cohérentes et innovantes.
Un étudiant dont la participation au rendu final projet est inexistante ne sera pas noté. Par ailleurs, ce projet nécessite d'y consacrer du temps, il n'est pas concevable de le réaliser la veille pour le lendemain. Vous devrez donc fournir un travail soutenu et de qualité dans les délais qui vous sont impartis. Un travail irrégulier au même qu'un travail insuffisant sera sanctionné. L'organisation du dépôt, sa propreté, la cohérence des `RELEASES` seront également prises en compte dans l'évaluation finale.

L'évaluation de la mise en oeuvre des concepts techniques s'appuiera sur une étude du code source et de la documentation technique. Le code est susceptible d'être exécuté pour s'assurer que les livrables fonctionnent bien. La vidéo et l'application déployée permettront quant à eux de valider la cohérence et le caractère innovant de votre application.

Un lien manquant ou une documentation incomplète entraîne systématiquement une perte de points importante puisqu'une partie du projet ne peut être évaluée correctement. Faites donc attention à ces aspects.

Un projet en retard sera sanctionné : 1 point / jour. Par ailleurs, la note du projet ne peut pas être compensée par un examen écrit (une consolidation).

[1] Auteur de ce paragraphe : Martin Quinson