

Laboratoire 1, partie 1 :

Création et configuration de l'environnement GitHub

Ce laboratoire a pour objectif de mettre en place l'environnement de développement GitHub qui sera utilisé pour supporter le travail de l'équipe pendant toute la session.

Dans ce laboratoire, vous devrez configurer l'environnement de travail GitHub de votre équipe et créer une structure de répertoires qui hébergera le code source des projets qui seront créés durant la session. Cet environnement GitHub sera utilisé pour le reste des laboratoires de la session.

Vous devrez aussi créer et configurer un projet, incluant un tableau Kanban, qui devra être utilisé tout au long de la session pour gérer (et documenter) la progression des tâches (Issues dans GitHub) associées aux divers travaux (dans le cadre des trois laboratoires) de l'équipe.

Prérequis

- Le laboratoire demande d'avoir un compte GitHub
- L'étudiant devra, pour les laboratoires, avoir des bases avec l'utilisation du système de contrôle de version **Git**.
- L'étudiant doit avoir une connaissance dans l'utilisation du langage de balisage [Markdown](#).

Files

StackEdit stores your files in your browser, which means all your files are automatically saved locally and are accessible **offline!**

Create files and folders

The file explorer is accessible using the button in left corner of the navigation bar. You can create a new file by clicking the **New file** button in the file explorer. You can also create folders by clicking the **New folder** button.

Figure 1 - Langage Markdown

Ressources

En suivant ce lien, vous trouverez de nombreuses ressources écrites et vidéos qui vous permettront de vous accompagner et de vous former au long de ces laboratoires : <https://itsemi.notion.site/LOG-680-32bf411835ff4566991ff11d1762c2b3>.

Les questions pertinentes qui seront posées par les étudiants seront à retrouver sur ce même lien afin de centraliser et de partager la connaissance.

Si vous avez quelconques recommandations, merci d'en faire part au chargé de laboratoires qui les ajoutera.

Lien du Discord afin d'échanger avec le chargé de laboratoire : <https://discord.gg/cdnydZnJ>.

Contenu

1. Création des répertoires GitHub

La première étape consiste à mettre en place deux répertoires pour y déposer le code source des projets.

Une personne de chaque équipe devra créer les repos GitHub et y ajouter ses coéquipiers et chargé de laboratoire. Le nom de votre repo GitHub doit minimalement contenir votre numéro de groupe et d'équipe (ex : **metrics-eq1** et **hvac-eq1**). Assurez-vous de créer des repos privés.

Vous pouvez aussi utiliser votre courriel de l'ETS pour obtenir un compte étudiant PRO gratuit. Les repos privés PRO permettent l'automatisation des politiques de branches, si cette fonctionnalité vous intéresse. Cependant, les repos basés sur une organisation (plutôt qu'un compte individuel) ne permettent pas l'automatisation de branche, même avec un compte PRO étudiant.

Vous êtes libre de choisir le type de repo (organisation ou individuel) que vous voulez du moment qu'il soit privé. Les comptes étudiants PRO ne sont pas obligatoires.

Néanmoins, je vous invite fortement à les utiliser car ils offrent de nombreux avantages, et cela en dehors dans la suite GitHub (Digital Ocean, Microsoft Azure, JetBrains, Namecheap...).

2. Création d'un projet et du tableau Kanban dans GitHub

Ensuite, vous devez créer un projet dans GitHub (voir onglet « Projects »). Ce projet vient avec un tableau Kanban. Celui-ci sera utilisé durant la session.

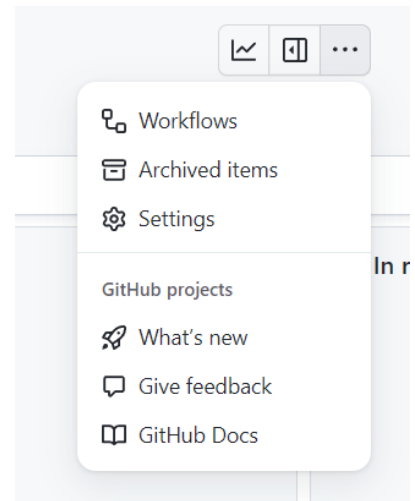
Le tableau doit comporter **au moins cinq colonnes** :

- Backlog
- À faire
- En cours
- Revue
- Terminée

Vous devez ajouter des “déclencheurs” (“trigger”) sur la colonne Backlog lors de l’ouverture de nouvelles tâches.

Vous devez aussi ajouter des “déclencheurs” (“trigger”) sur la colonne Terminée lors de la fermeture de tâches.

GitHub continuant d’évoluer avec son temps, les déclencheurs sont maintenant remplacés par des workflows plus visuels que vous trouverez directement dans vos projets.



3. Ajout de modèles et d’étiquettes

Pour le projet, vous devrez ajouter des modèles ("template") pour les tâches, des modèles pour les « Pull Requests », et des étiquettes (labels) pour les tâches.

Le projet devra comporter **au moins deux modèles pour les tâches et un modèle pour les « Pull Requests »**. Le projet doit aussi avoir **un nombre approprié d’étiquettes**. Il est possible de les utiliser pour définir des priorités, des types de tâches, etc. **Vous devez justifier vos choix** pour que les modèles et étiquettes soient pertinents au projet.

4. Création de tâches

À partir de maintenant, vous devez utiliser votre Kanban pour la gestion de votre projet. Pour ce faire, vous devez créer une tâche (“issue” dans GitHub) dans votre Kanban pour chacune des tâches de votre de projet. Vous devez utiliser les modèles ajoutés au préalable. Bien que les tâches **doivent respecter le modèle utilisé**, il n’est pas nécessaire de mettre une description exhaustive pour celle-ci.

5. Politique de branches

Vous devez adopter une politique de branches pour votre repo GitHub. Nous suggérons l’utilisation de politique de branches connues dans l’industrie :

<https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow>

<https://nvie.com/posts/a-successful-git-branching-model>

<https://guides.github.com/introduction/flow/>

La structure de branche doit être respecté avec “main”, “develop” et “features/*” (avec git flow) ou “main” et “feature” (avec github flow).

De plus, la documentation devrait être mise à jour pour refléter les changements au

projet, par exemple indiquer comment exécuter le code, justifier les métriques sélectionnées, etc.

6. Création de documentation

Vous devrez documenter les différents aspects de votre projet durant la session. Pour ce faire, vous pouvez utiliser le fichier README.md du projet et créer un répertoire appelé *Wiki* contenant la documentation dans différents fichiers en format Markdown. Ces fichiers doivent contenir toutes les diverses informations pertinentes au projet.

Pour le laboratoire 1 (Partie 1 et 2), vous devez y mettre minimalement le rapport ainsi qu'une page principale comprenant une petite description du projet.

Consignes additionnelles

- La documentation faite par les membres du groupe doit être :
 - Claire
 - Concise
 - Pertinente
 - Simple.
- Il est **très important d'utiliser les jalons (Milestones) pour chacune des tâches (et/ou « Pull Requests ») des laboratoires** (ex. Lab1). L'utilisation de ceux-ci simplifiera grandement les corrections puisqu'il sera possible de filtrer les tâches à partir de ce paramètre.
- Bien que GitHub propose une liste (générique) de "templates" et "labels", ceux-ci ne sont pas tous pertinents aux besoins spécifiques de votre projet. Vous devez créer une liste de "templates" et "labels" qui sont adaptés aux besoins de votre projet, qui peut inclure certains proposés par GitHub.
- Votre application doit être facile à utiliser et configurer
- Imaginez que votre projet est *OpenSource* et qu'une nouvelle personne peut s'intégrer au projet facilement. Tout doit être documenté dans les fichiers Markdown du *Wiki* : comment exécuter le projet, libellé, modèles, structure de branche, etc.

Rapports

Il y a un seul rapport à rendre sur Moodle pour le TP1 (partie 1 et 2). Vous devez soumettre un document qui comprend une introduction, une description du projet, une explication des décisions (ex. justification du choix des métriques, comment chaque métrique est calculée, etc.) et une conclusion. Le rapport peut faire référence à votre wiki / documentation dans GitHub.