

Versionner un projet avec Git, Communiquer avec GitHub

Florian Legendre

Université de Poitiers

Année 2020 - 2021

Table of Contents

- 1 Pourquoi GitHub ?
- 2 Communiquer le projet avec GitHub
- 3 Communiquer autour du projet avec GitHub

Table of Contents

1 Pourquoi GitHub ?

- Les contraintes à respecter
- Un extrait de l'état de l'art
- Les motivations de ce choix

Les contraintes à respecter

Peu importe la plateforme choisie nous devons respecter les deux contraintes suivantes :

- Les données/codes ne doivent pas être accessibles au publique ou à qui que ce soit d'autres que les personnes autorisées
- Les données/codes doivent n'appartenir qu'à vous en tout temps

Les autres solutions

SourceSup RENATER :

Analyse de la solution SourceSup Renater	
AVANTAGES	INCONVÉNIENTS
<ul style="list-style-type: none">• Service de dépôt distant• Création de forums autour du projet• Documentation du projet par un Wiki / Page web• Serveur de fichiers lourds (300Mo)• Listes de diffusions / Sondages / Abonnements à des forums• Messagerie intégrée• Sécurité des données / Pas de commercialisation des données• Des mots-clés pour les recherches de projets• Extensibilité des services grâce aux plugins	<ul style="list-style-type: none">• Boutons et liens parfois très petits voire illisibles• Style de l'interface un peu vieillissant... Peut ne pas être engageant.• Pas d'intégrations à d'autres plateformes comme Zenado ou Gitter, etc.• Validation des projets déposée un peu floue• Réactivité du support ? Survie de la plateforme ?• Certaines fonctionnalités peuvent être très techniques

En-dehors de SourceSup RENATER les solutions étaient soit :

- Partielles (GitLab / Bitbucket)
- Payantes (Bitbucket)
- Les termes d'utilisations étaient flous (GitLab)
- Trop techniques à mettre en oeuvre (Allura)

Si vous voulez tout de même les explorer voici quelques liens :

- SourceSup RENATER : <https://sourcesup.renater.fr/>
- GitLab : <https://about.gitlab.com/>
- BitBucket : <https://bitbucket.org/product/>
- Allura : <https://allura.apache.org/>

Avantages et Inconvénients de GitHub

Analyse de GitHub	
AVANTAGES	INCONVÉNIENTS
<ul style="list-style-type: none">● Respect des contraintes énoncées par le biais des projets privés● Nombre de collaborateurs et de projets privés illimité + de nombreux services gratuits (cf. https://github.com/pricing)● Tout-en-un pour la communication/gestion (humaine) des projets● Possibilité de créer des organisations● La forge n°1 sur le marché => plateforme vivante et dynamique	<ul style="list-style-type: none">● Forge propriétaire (attention à d'éventuels changements des conditions d'utilisation...)● Quelques limitations sur les projets privés avec la formule gratuite (pas de Wiki ni de pages web par exemple)

Table of Contents

- 2 Communiquer le projet avec GitHub
 - Créer un dépôt et régler sa visibilité
 - Collaborer

Régler la visibilité à la création du dépôt

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner ^{*} Repository name ^{*}

Churclub /

Great repository names are short and memorable. Need inspiration? How about [potential-eureka?](#)

Description (optional)

☒ **Public**
Anyone on the Internet can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

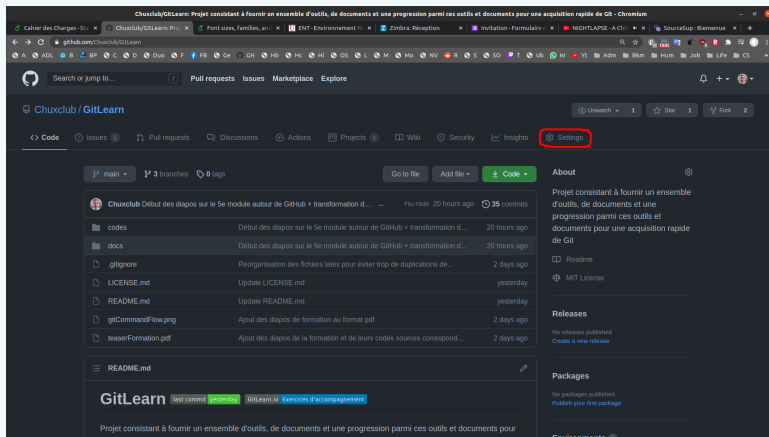
☒ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

☐ **Add .gitignore**
Choose which files not to track from a list of templates. [Learn more.](#)

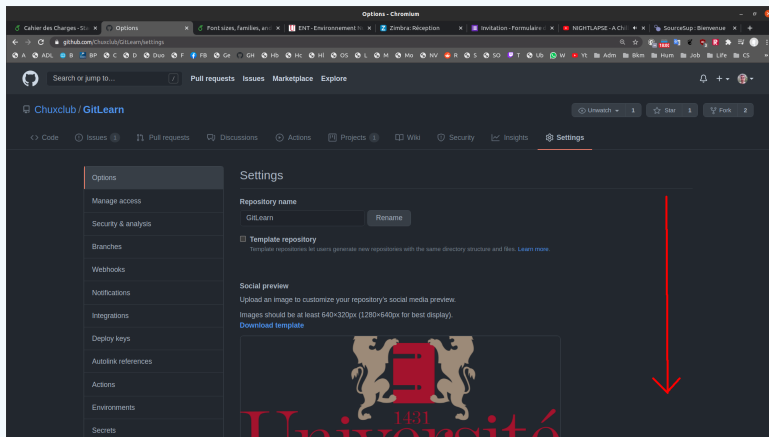
☐ **Choose a license**
A license tells others what they can and can't do with your code. [Learn more.](#)

Create repository

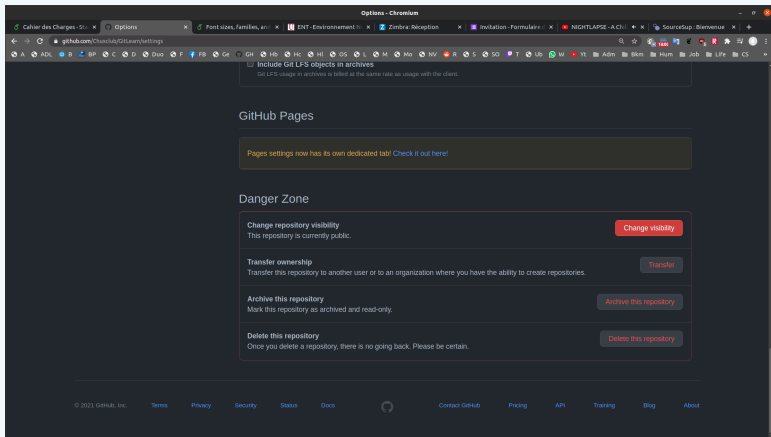
Régler la visibilité après la création du dépôt



Régler la visibilité après la création du dépôt



Régler la visibilité après la création du dépôt



Inviter un collaborateur

Il y a trois scénarios possibles, qui dépendent de la configuration du ou des dépôts distants.

Premier scénario :

Vous êtes indiqué comme **collaborateur** sur la branche distante. En tant que collaborateur vous pouvez "pusher" les changements que vous voulez sans restriction ! Sauf si le projet fait partie d'une organisation...

ATTENTION pour les chefs de projet : Cela nécessite d'avoir une très grande confiance en ses collaborateurs. En cas de doute une des solutions est alors d'avoir une branche distante de développement

Inviter un collaborateur

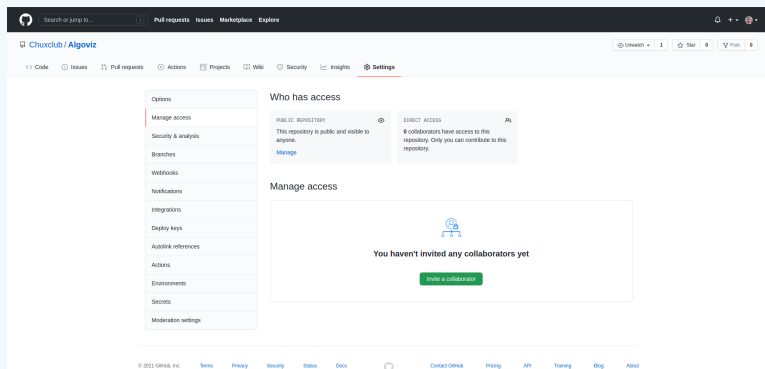


Figure – Définir des collaborateurs sous GitHub

Fork et Pull request

Second scénario :

- 1 Vous avez "**forké**" (càd copié) sous Github le projet d'une personne.
- 2 Vous avez ensuite cloné ce fork.
- 3 Vous avez travaillé sur ce clone et "pushé" vos projets sur votre fork distant.
- 4 Vous pouvez ensuite faire un pull request directement sur l'interface GitHub.

Fork et Pull request

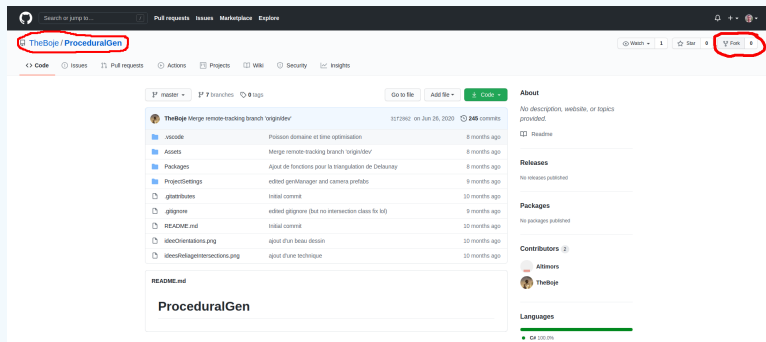


Figure – Forker sous GitHub

Fork et Pull request

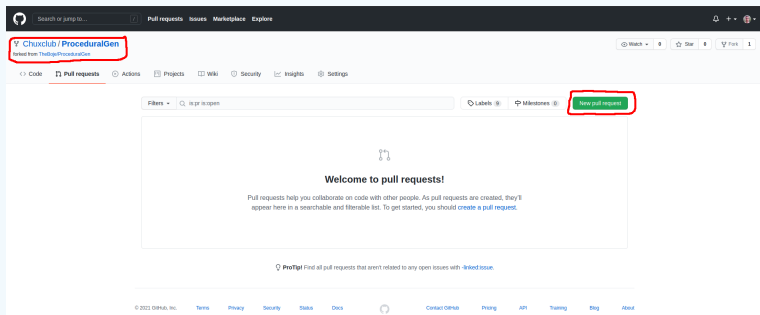


Figure – Faire un pull request sous GitHub

Fork tardif

Dernier scénario :

Vous avez **cloné** le projet d'une personne. Vous avez ensuite travaillé dessus et vous vous rendez compte que vous souhaiteriez faire part de vos modifications à l'auteur...

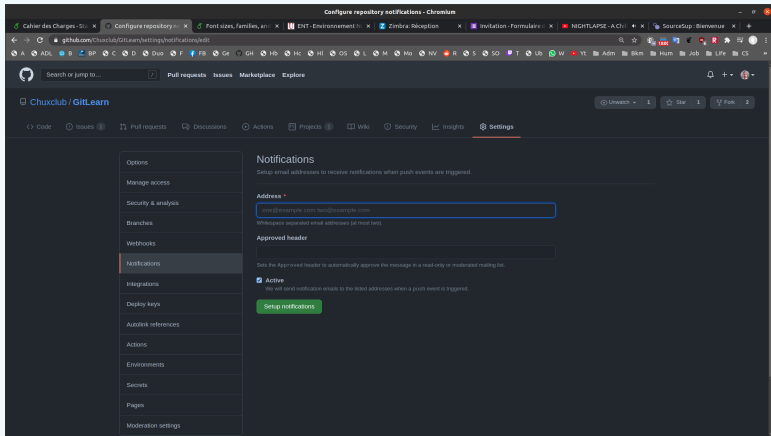
- ❶ Comme précédemment vous forcez
- ❷ Vous ajoutez une branche distante à votre clone : `git remote add <nom> <urlDeVotreDepot>` ou, plus brutal : `git remote rm origin` suivi de `git remote add origin <urlDeVotreDepot>`
- ❸ Vous poussez vos changements sur ce fork
- ❹ Vous faites un pull request comme précédemment

Table of Contents

- 3 Communiquer autour du projet avec GitHub
 - Communiquer
 - Les fichiers spéciaux
 - La syntaxe Markdown

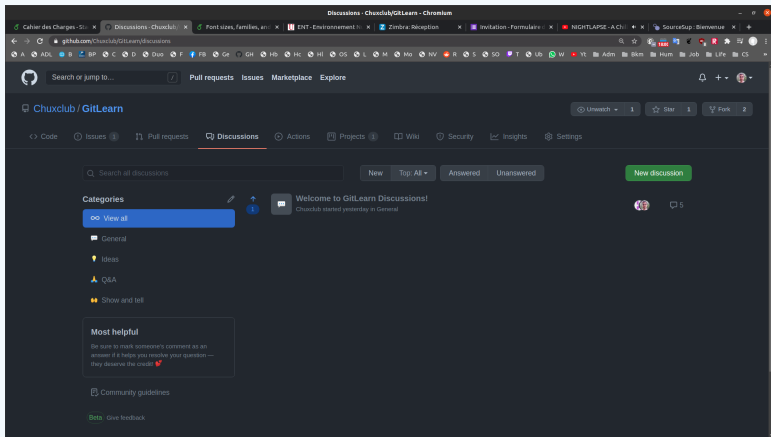
Les notifications

Les notifications permettent à celles et ceux qui y sont inscrits de recevoir des mails dès qu'une mise à jour est faite sur le projet :



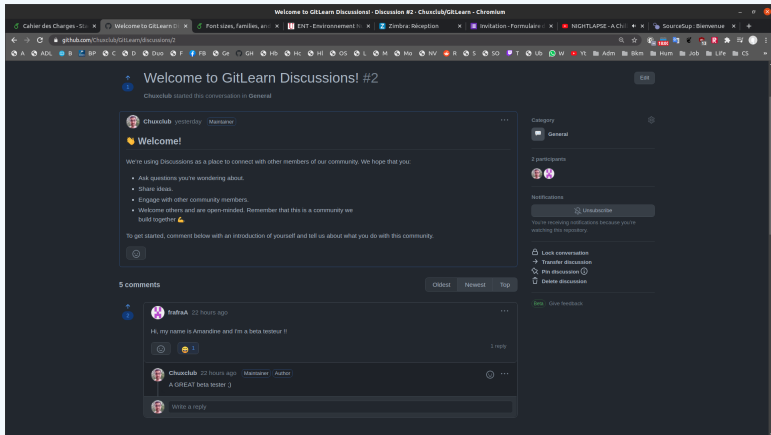
Les discussions

Les discussions sont des sortes de listes de diffusion de mails avec des sujets fixes :



Les discussions

Il est possible de désabonner d'une discussion pour cesser de recevoir des mails ou au contraire d'épingler celle-ci pour qu'elle reste visible à tout moment :



Les discussions

Il est possible d'y répondre directement depuis sa boîte mail :

The screenshot shows the Thunderbird email client interface. The left sidebar displays the folder structure, including 'Inbox - florian-1992@hotmail.fr'. The main pane shows a list of emails in the inbox. The selected email is from 'fratrac-notifications@github.com' with the subject 'Re: [Chuxclub/GitLearn] Welcome to GitLearn Discussions! (#2)'. The email body contains a warning about privacy and a link to download Outlook for Android.

From: fratrac-notifications@github.com
Subject: Re: [Chuxclub/GitLearn] Welcome to GitLearn Discussions! (#2)
Reply to: Chuxclub/GitLearn <reply@LIST.JDKS.IZMCHCLFYXBGHVFYVYNNHVBMMXAAHJWZ@reply.github.com>
To: Chuxclub/GitLearn <GtLearn@noreply.github.com>
From: Me <florian-1992@hotmail.fr>
Subject: Re: [Chuxclub/GitLearn] Welcome to GitLearn Discussions! (#2)

To protect your privacy, Thunderbird has blocked remote content in this message.

L'avalanche de mails !!

Télécharger Outlook pour Android <https://aka.ms/AAb9ysg>

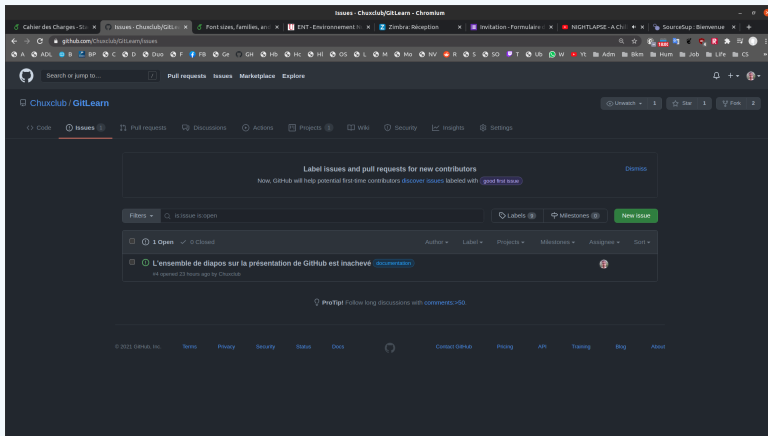
From: Florian Legendre ***@***<***>
 Sent: Thursday, April 21, 2021 5:42:44 PM
 To: Chuxclub/GitLearn ***@***<***>
 Cc: fratrac ***@***<***>; Comment ***@***<***>
 Subject: Re: [Chuxclub/GitLearn] Welcome to GitLearn Discussions! (#2)



1431
Université
de Poitiers

Les tickets (ou issues)

Un ticket est un moyen d'indiquer qu'une feature est souhaitée ou un bug a été détecté ou même tout simplement de poser une question.

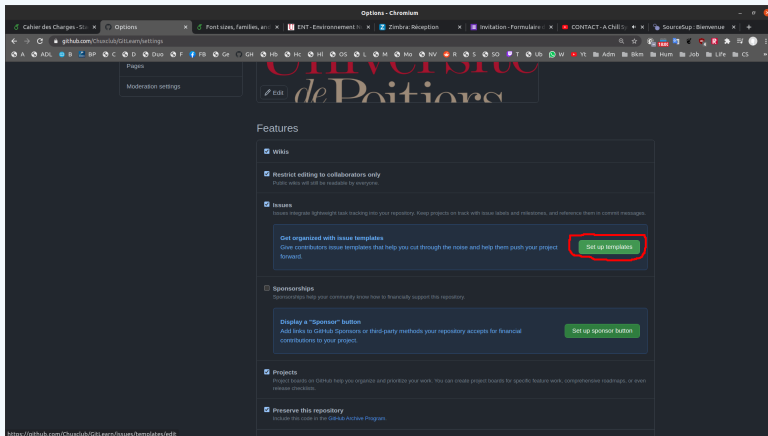


Les "billets" ou "tickets" présentent plusieurs avantages :

- Ils peuvent être catégorisés par des labels et triés automatiquement dans un tableau de bord d'avancement du projet donné (cf. "Les projets" plus loin)
- La personne à qui un ticket est assigné reçoit un mail de son assignation avec le message du ticket (comme pour les discussions on peut y répondre directement par mail!)
- On peut faire des "templates" de tickets ce qui permet une prise en main claire et rapide pour la personne assignée au ticket
- Les tickets gardent un historique des bugs éliminés, ce qui permet de savoir quand un bug est revenu ou non

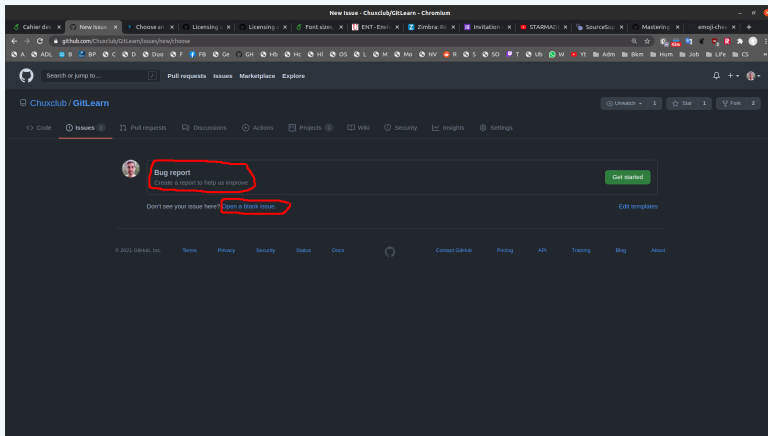
Les tickets (ou issues)

Pour créer un template de ticket il faut d'aller dans les options du dépôt :



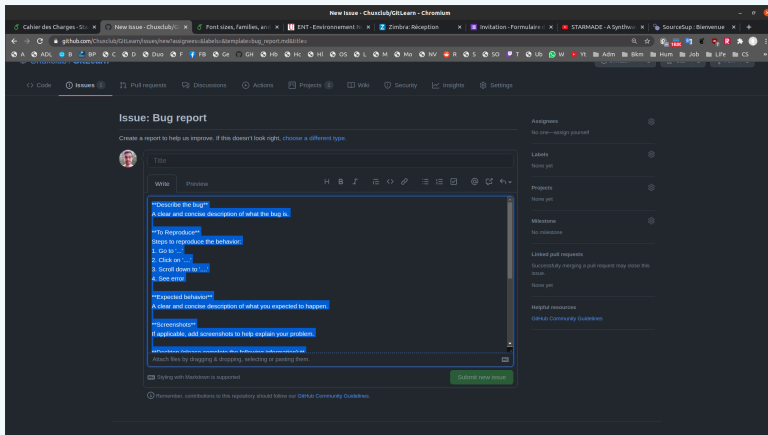
Les tickets (ou issues)

L'utilisateur peut alors choisir de créer un billet en utilisant le template ou, si aucun template ne lui convient, il peut choisir de créer un billet sans template :



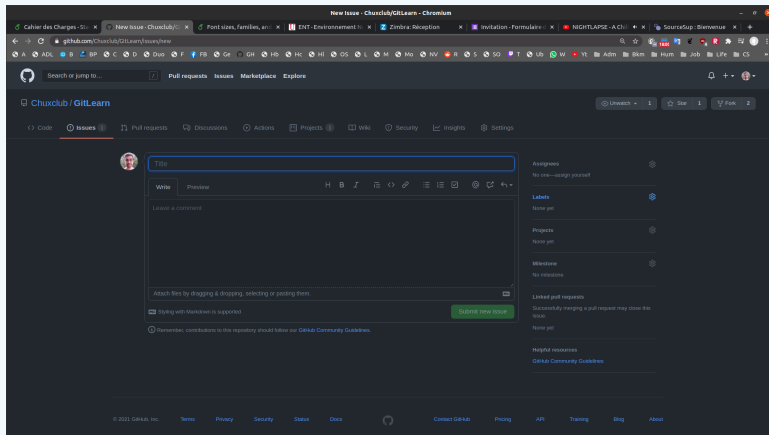
Les tickets (ou issues)

Édition d'un billet avec template :



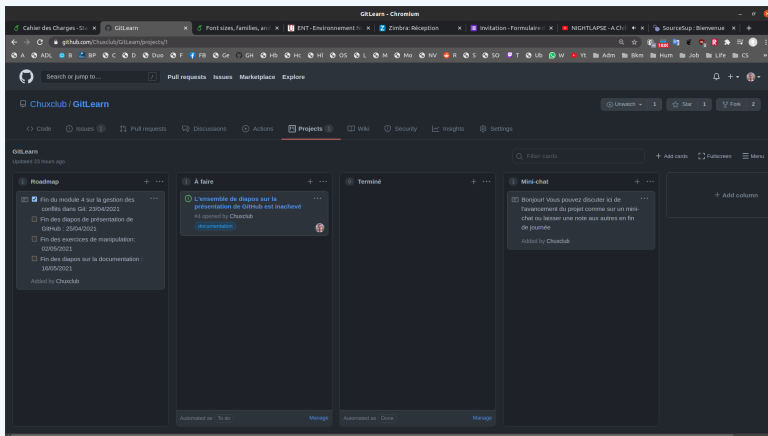
Les tickets (ou issues)

Édition d'un billet sans template :



Les projets

Les projets sont des tableaux de bord de l'avancement du projet. Les tickets peuvent y apparaître automatiquement et triés en colonnes "à faire" / "en cours" / "terminés" :

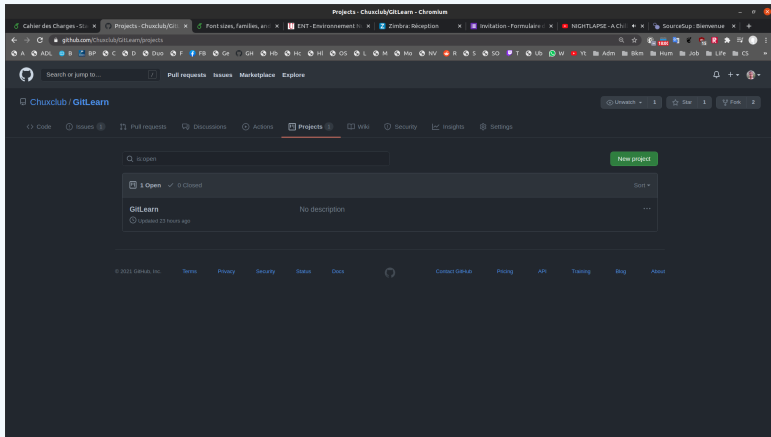


Les projets fonctionnent en utilisant deux briques : les colonnes et dans les colonnes des cartes. Il est possible d'utiliser la syntaxe Markdown dans ces cartes. Aussi vous pouvez donner le sens et l'usage que vous voulez à vos colonnes :

- Créer une roadmap du projet
- Un mini-chat
- Une colonne de liens vers des documentations extérieures
- Les gestions de tickets
- Une colonne "dates importantes" qui donne les deadlines ou la date de la prochaine réunion autour du projet
- etc.

Les projets

Il est possible de créer plusieurs projets dans les cas où, par exemple, le projet implique plusieurs gros services différents :



Le fichier README

Chuxclub/GitLearn: Projet consistant à fournir un ensemble d'outils, de documents et une progression parmi ces outils et documents pour une acquisition rapide de Git - Chromium

Chuxclub/GitLearn

main branches tags

Go to file Add file Create

About

Projet consistant à fournir un ensemble d'outils, de documents et une progression parmi ces outils et documents pour une acquisition rapide de Git

chuxclub.github.io/gitlearn/

github clone ce projet

Readme

MIT License

Environnements

github-pages Active

Languages

100.0%

Chuxclub Début des diapos sur le Se module autour de GitHub + transformation d... 21 hours ago 16 commits

File	Commit	Time
codes	Début des diapos sur le Se module autour de GitHub + transformation d...	21 hours ago
docs	Début des diapos sur le Se module autour de GitHub + transformation d...	21 hours ago
.gitignore	Réorganisation des fichiers latex pour éviter trop de duplications de...	2 days ago
LICENSE.md	Update LICENSE.md	yesterday
README.md	Update README.md	yesterday
gitCommandFlow.png	Ajout des diapos de formation au format pdf	2 days ago
leaerFormation.pdf	Ajout des diapos de la formation et de leurs codes sources correspond...	2 days ago

README.md

GitLearn [last commit](#) [yesterday](#) [GitLearn.io](#) [Exercices d'accompagnement](#)

Projet consistant à fournir un ensemble d'outils, de documents et une progression parmi ces outils et documents pour une acquisition rapide de Git

Extrait de l'arborescence du projet 🌿

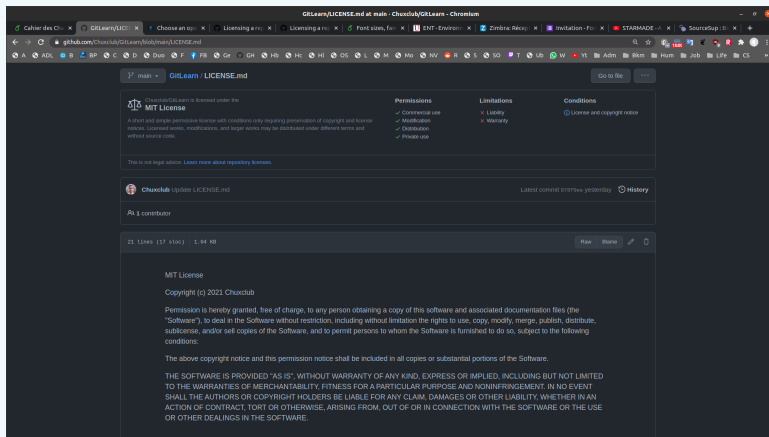
```

- codes
  - test.txt
- docs
  - diapos_formation
    - 9-3_interludeTernauxCad.pdf
    - 9-3.pdf
  - LICENCE.md
  - .gitignore
  
```

<https://github.com/Chuxclub/GitLearn/blob/main/LICENCE.md>

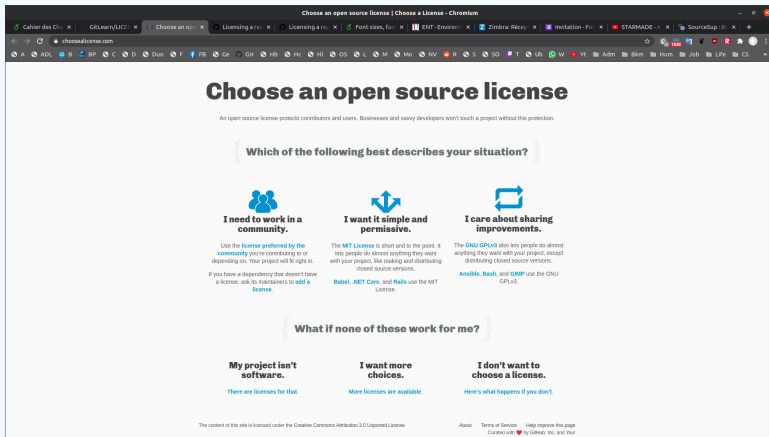
Le fichier LICENSE

À l'instar du README le fichier LICENSE est analysé par un robot pour afficher cette licence dans la page d'accueil de votre projet :



Le fichier LICENSE

Le contenu du fichier LICENSE peut être copié-collé en vous aidant de ce site <https://choosealicense.com/> :



Le guide du Markdown

Le Markdown est un ensemble de petits codes syntaxiques qui permettent de formater automatiquement un document.

L'ensemble de ces codes tient en une page web <https://guides.github.com/features/mastering-markdown/> :

[//guides.github.com/features/mastering-markdown/](https://guides.github.com/features/mastering-markdown/) :

