

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
- 4 Pour aller plus loin 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/>

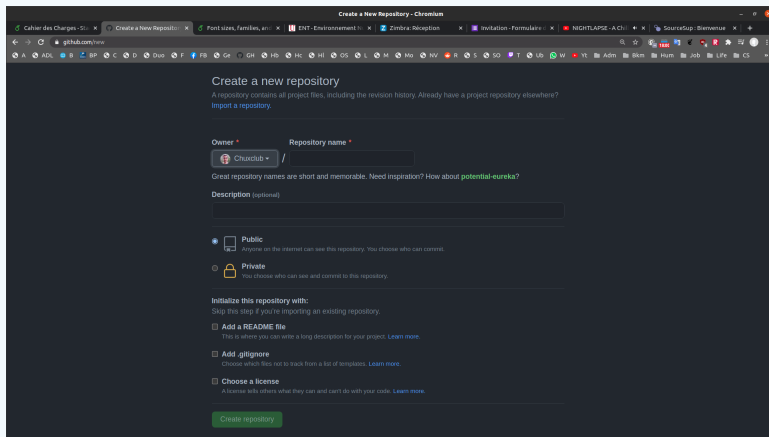
Avantages et Inconvénients de GitHub

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.

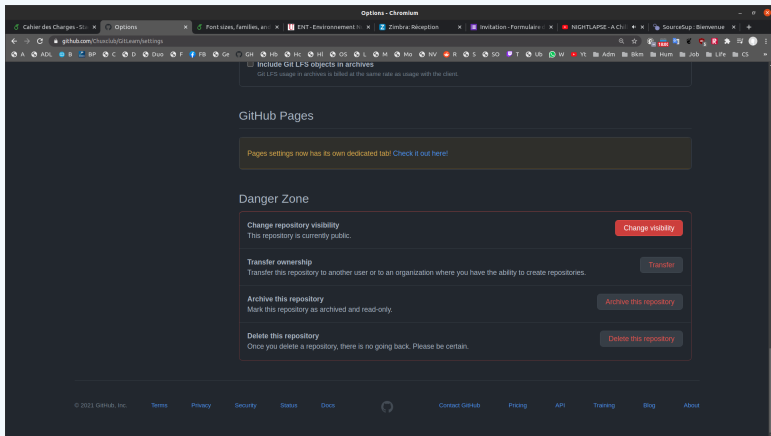
☒ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

☒ **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



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

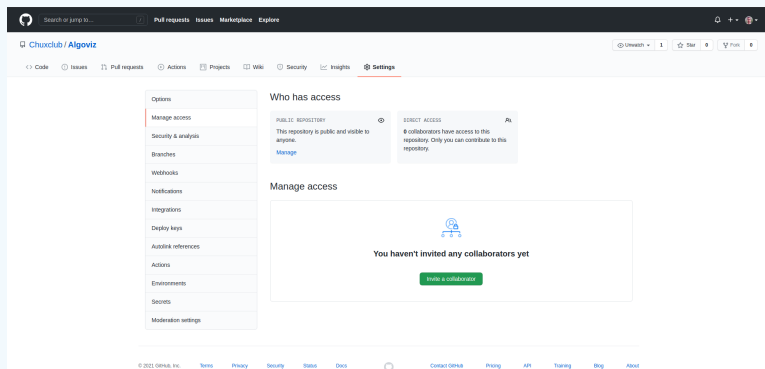


Figure – Définir des collaborateurs sous GitHub

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

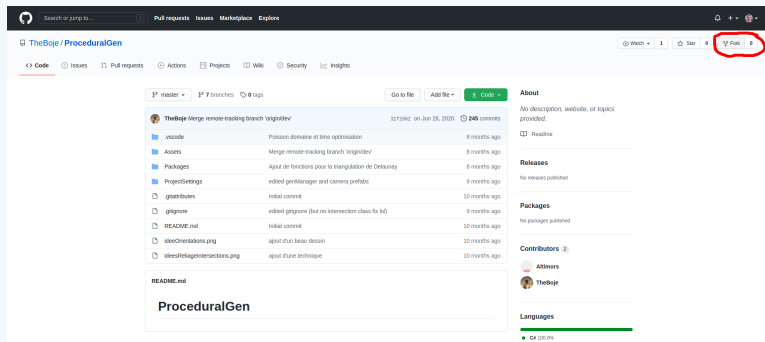


Figure – Forker sous GitHub

Fork et Pull request

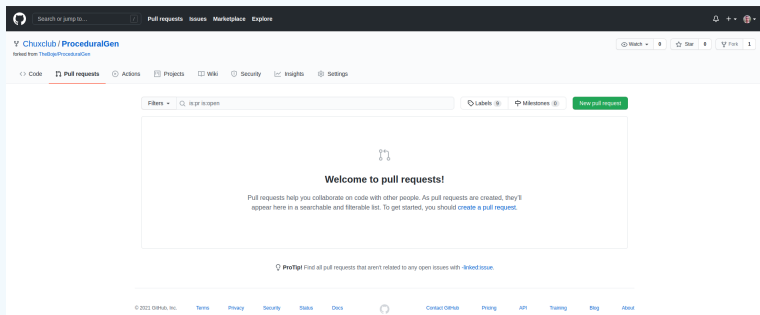


Figure – Faire un pull request sous GitHub

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...

- 1 Comme précédemment vous forcez
- 2 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>"
- 3 Vous poussez vos changements sur ce fork
- 4 Vous faites un pull request comme précédemment

Table of Contents

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

Table of Contents

- 4 Pour aller plus loin avec GitHub
 - Le README de son profil
 - Les Shields en tête du README
 - Les organisations
 - Les logiciels avec une API GitHub

