

# ***TODO-LIST***

Procedure de collaboration

## Section 1 : Collaborer au projet d'amélioration

Ce document est destiné à toutes les personnes rejoignant le projet, afin de comprendre comment intégrer le projet en toute harmonie.

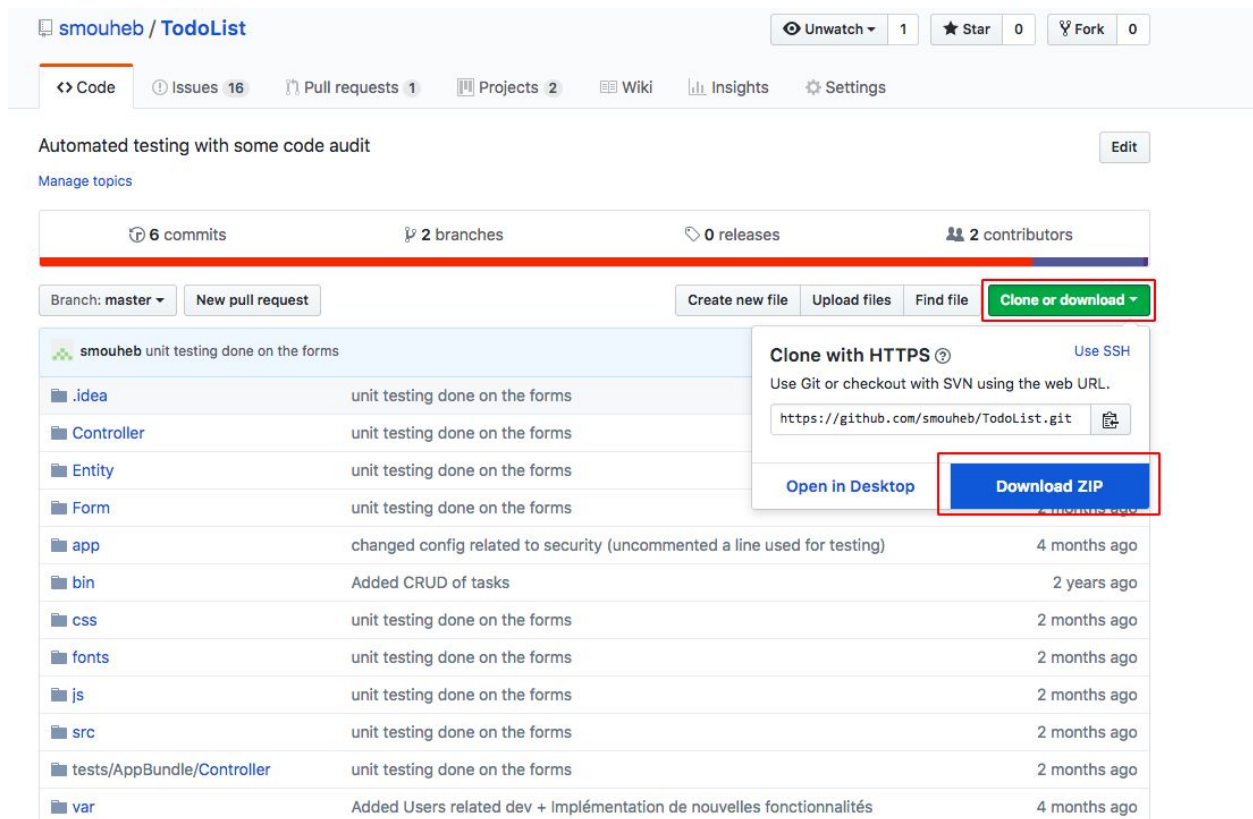
- 1.1 Cloner et installer le projet
- 1.2 Choisir un item dans le backlog
- 1.3 Règle avant de faire un pull request
- 1.4 Pull request et convention de nommage

## 1.1 Cloner et installer le projet

**La première chose à faire est de cloner/ télécharger le projet en local.**

### 1.1.1 Téléchargement du projet par dossier

En utilisant l'interface utilisateur, nous avons la possibilité de télécharger un dossier compresser en pressant le bouton indiqué ci-dessous.



smouheb / TodoList

Unwatch 1 Star 0 Fork 0

Code Issues 16 Pull requests 1 Projects 2 Wiki Insights Settings

Automated testing with some code audit [Manage topics](#)

6 commits 2 branches 0 releases 2 contributors

Branch: master New pull request Create new file Upload files Find file Clone or download

smouheb unit testing done on the forms

.idea	unit testing done on the forms	
Controller	unit testing done on the forms	
Entity	unit testing done on the forms	
Form	unit testing done on the forms	
app	changed config related to security (uncommented a line used for testing)	4 months ago
bin	Added CRUD of tasks	2 years ago
css	unit testing done on the forms	2 months ago
fonts	unit testing done on the forms	2 months ago
js	unit testing done on the forms	2 months ago
src	unit testing done on the forms	2 months ago
tests/AppBundle/Controller	unit testing done on the forms	2 months ago
var	Added Users related dev + Implémentation de nouvelles fonctionnalités	4 months ago

Clone with HTTPS Use SSH

Use Git or checkout with SVN using the web URL.

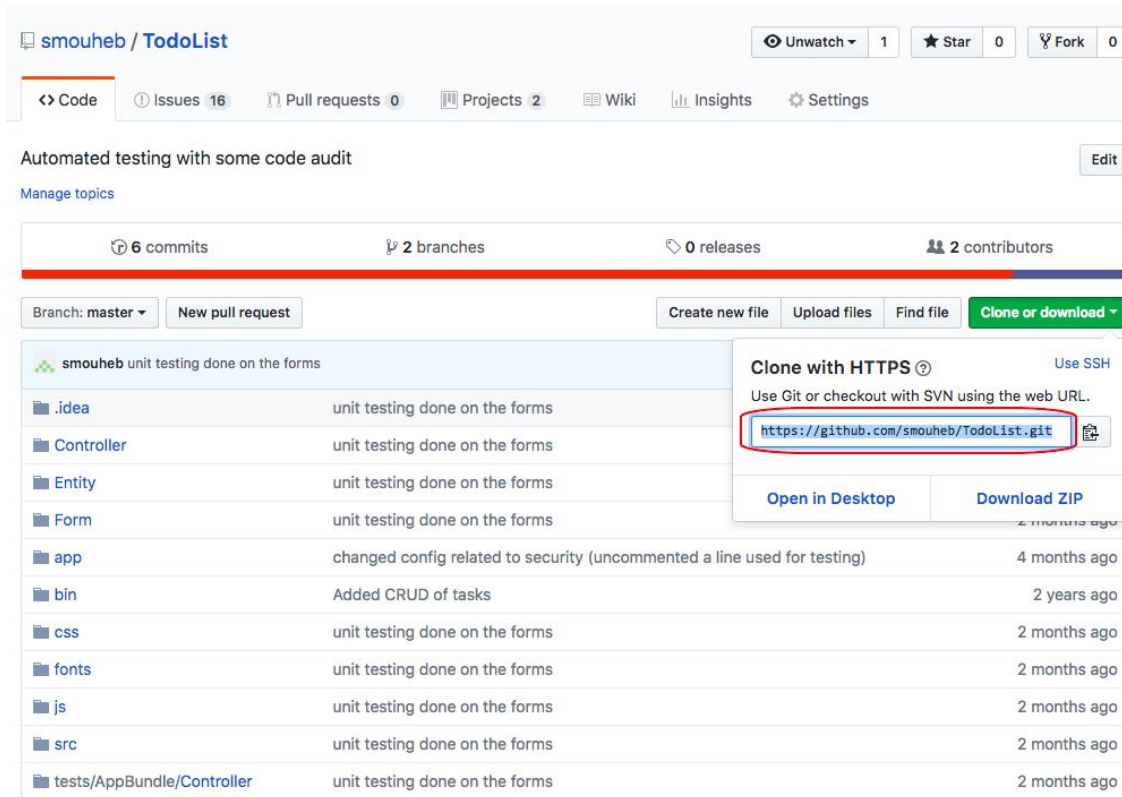
<https://github.com/smouheb/TodoList.git>

Open in Desktop Download ZIP

## 1.1.2 En Utilisant l'invite de commande

Ouvrez votre terminal et lancer la commande ci-dessous:

“*Git clone* <https://github.com/smouheb/ToDoList.git>”



The screenshot displays the GitHub repository page for `smouheb / ToDoList`. The repository has 16 issues, 0 pull requests, 2 projects, 0 stars, and 0 forks. The main content area shows a list of files and folders, including `.idea`, `Controller`, `Entity`, `Form`, `app`, `bin`, `css`, `fonts`, `js`, `src`, and `tests/AppBundle/Controller`. A modal is open, showing the 'Clone with HTTPS' option, with the URL `https://github.com/smouheb/ToDoList.git` highlighted.

File/Folder	Description	Last Commit
<code>.idea</code>	unit testing done on the forms	2 months ago
<code>Controller</code>	unit testing done on the forms	2 months ago
<code>Entity</code>	unit testing done on the forms	2 months ago
<code>Form</code>	unit testing done on the forms	2 months ago
<code>app</code>	changed config related to security (uncommented a line used for testing)	4 months ago
<code>bin</code>	Added CRUD of tasks	2 years ago
<code>css</code>	unit testing done on the forms	2 months ago
<code>fonts</code>	unit testing done on the forms	2 months ago
<code>js</code>	unit testing done on the forms	2 months ago
<code>src</code>	unit testing done on the forms	2 months ago
<code>tests/AppBundle/Controller</code>	unit testing done on the forms	2 months ago

### 1.1.3 Installation avec Composer

Ouvrez votre terminal et allez dans le répertoire où le projet est sauvegardé.

Lancer l'installation à l'aide de la commande suivante:

*“Composer install”*

Ensuite entrer les détails de votre base de données (host, port, username etc...)

Et pour finir lancer la commande afin de créer votre base de données et/ou le schéma (lancer une migration si vous utilisez cette fonctionnalité de doctrine)

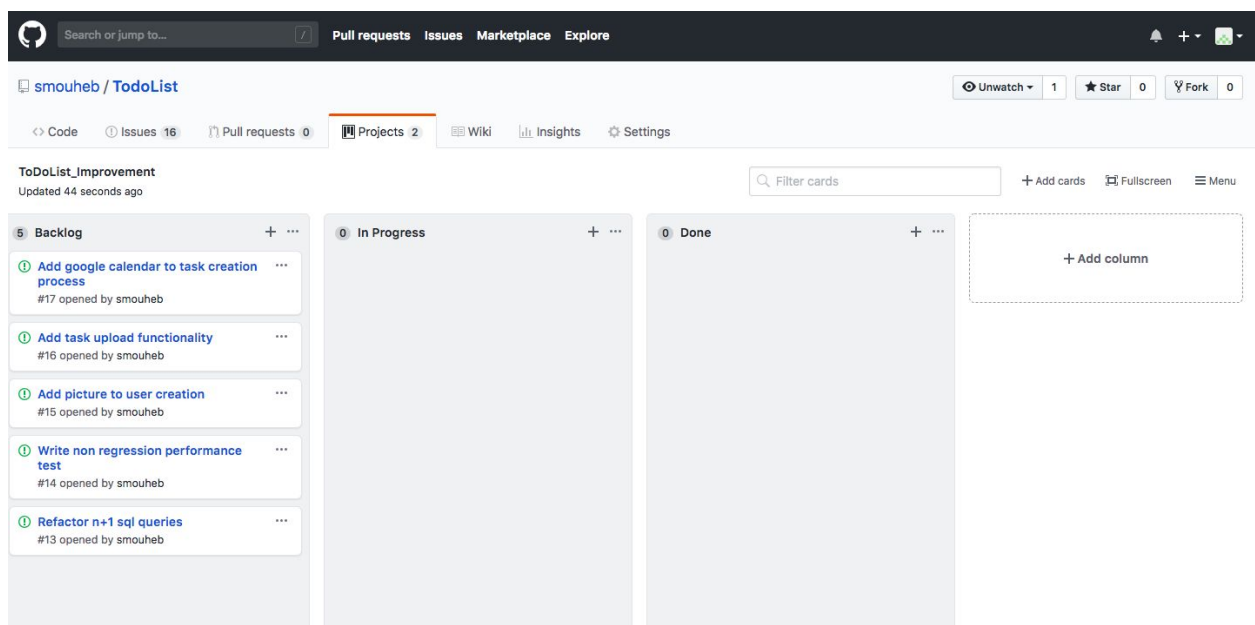
*bin/console doctrine:database:create*

*bin/console doctrine:schema:create*

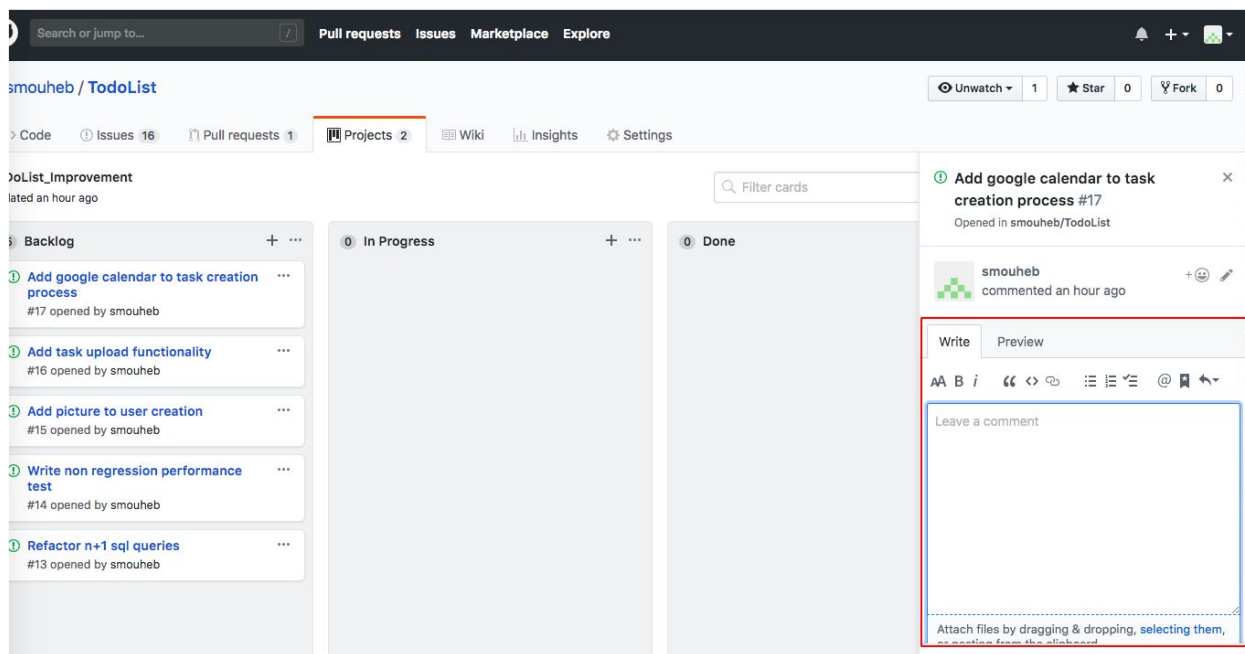
## 1.2 Choisir un item dans le backlog

A l'aide de la section précédente votre projet est maintenant installé en local, vous pouvez commencer à découvrir le code source de cette application.

Afin de pouvoir rentrer rapidement dans le vif du sujet, un projet a été créé sur le repository de l'application listant toutes les “issues” dans un backlog comme montré ci-dessous



Chaque ticket sera au préalable assigné et la personne en question pourra commenter le ticket si besoin éditant l'issue:



Une fois l'issue prise en charge par le développeur, le collaborateur devra placé le ticket dans la colonne in progress afin the tracker l'évolution du projet.

### 1.3 Règle avant de faire un pull request

Une fois le ticket résolu, le développeur devra s'assurer de commiter les points suivants avec le code:

- Profiling the l'application avant et après le développement
- Rapport en pdf de retour de test de non régression (en utilisant les tests existants)
- Rapport en pdf de retour de test implémenté pour les nouvelles fonctionnalité
- Rapport en pdf de la revue du code (bonnes pratiques etc...) avec codacy

Une fois tout cela préparer, lancer le pull request en suivant les indications mentionnés au point suivant.

## 1.4 Pull request et convention de nommage

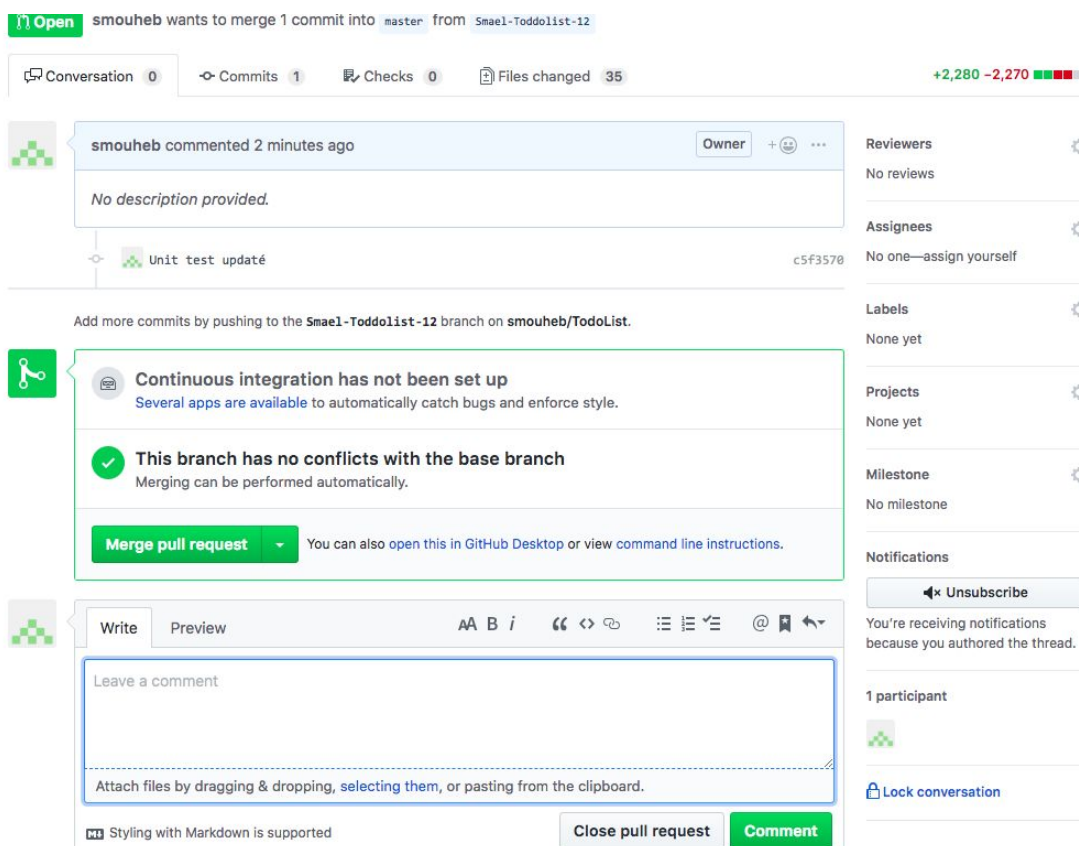
Pour lancer les pull request:

Tous les commit lors du développement devront être bien décrits (-m "xxxxx")

- ***Git commit -a -m"xxxxx" && git push origin master:<Nom Du Dev + projet + Id de l'issue>***

Comme montré ci-dessus, la convention de nommage pour chaque commit sur le remote repository devra suivre ce qui est indiqué ex: **git push origin master:Smael-ToDoList-17**

- Pour lancer un pull request veuillez indiquer les éléments nécessaire à la vérification



**Note:** même en local il est conseillé de créer des branches et faite des “merge” au fur et à mesure du développement.