Démo TEST REACT et Le lié à Travis CI

Avant de commencer notre test :

1. Ouvrir un GITBASH du projet (clique droite sur le projet REACT et clique sur GITBASH, si tu n’as pas téléchargé GITBASH, aller sur google pour le télécharger)
2. Dans GITBASH, écrire la command « npm install --save-dev @testing-library/react » pour avoir l’accès sur les tests.

Après l’installation du « @testing-library » :

1 - créer un nouveau dossier dans le dossier « src/components » du projet et le nommé « test » (clique droite sur le src/components et clique sur nouveau dossier)

2 - créer un fichier dans le dossier « test » (clique droite sur le dossier « test » et clique sur nouveau fichier) et le nommés avec le nom du test que vous voulez tester, et à la fin ajouter « .test.js » (exemple : je veux tester mon buton, donc je lui nomme « button.test.js »)

1. importer les fichiers que t’as besoin.

(import React from 'react';

import ReactDom from 'react-dom';

import { render, cleanup } from '@testing-library/react';

import "@testing-library/jest-dom/extend-expect";)

Aussi importer le fichier que tu veux tester

Exp : (import Button from 'le chemin de fichiers';)

4 -ajouter un « afterEach(cleanup); » à la fin de ton dernier import (il serre à nettoyer tes traces de test)

5- écrire

it("le nom de ton test", () => {

« les test»

})

exp :

it("test label du button", () => {

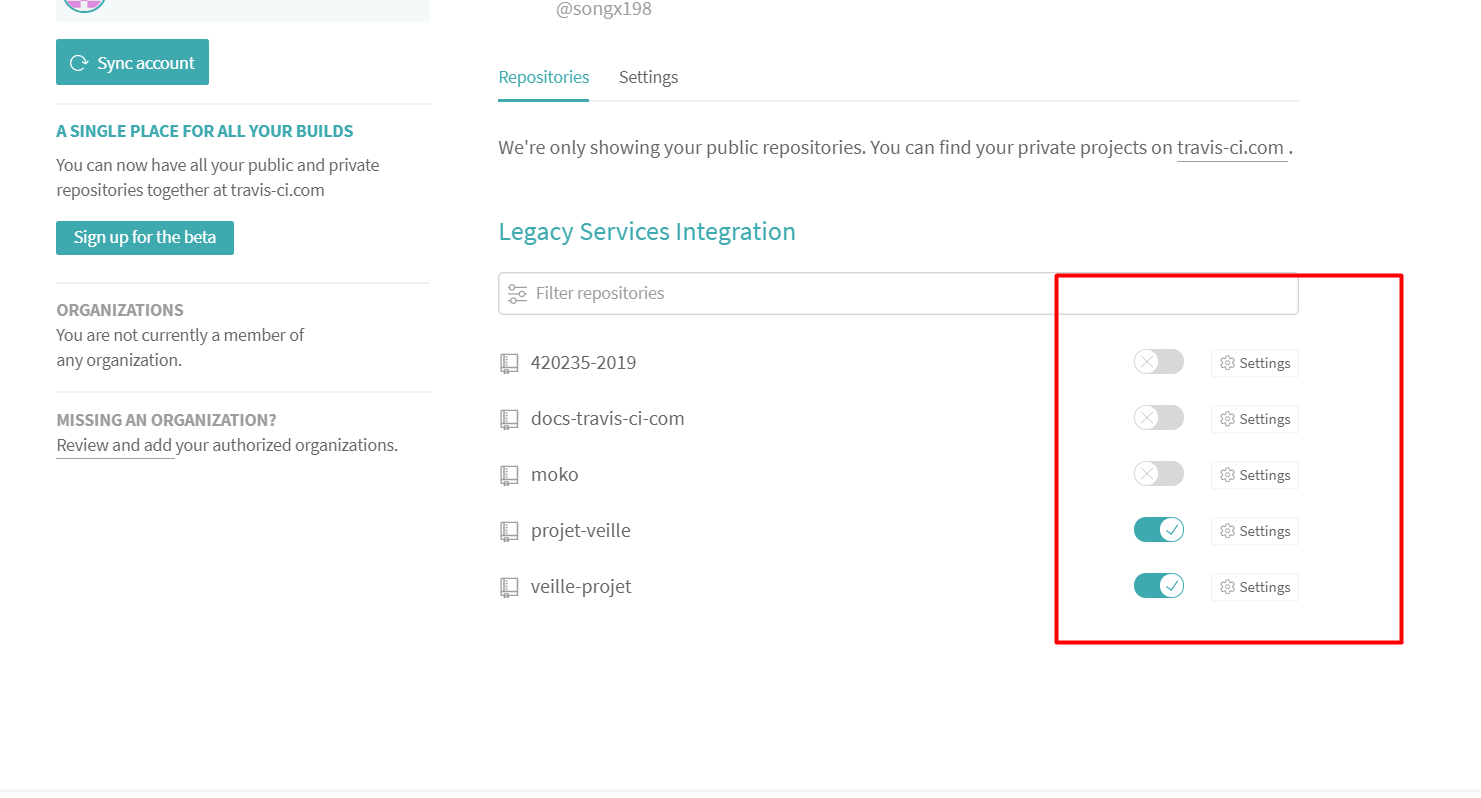
const { getByTestId } = render(<Button label="Click"></Button>)

expect(getByTestId('button')).toHaveTextContent("Click")

})

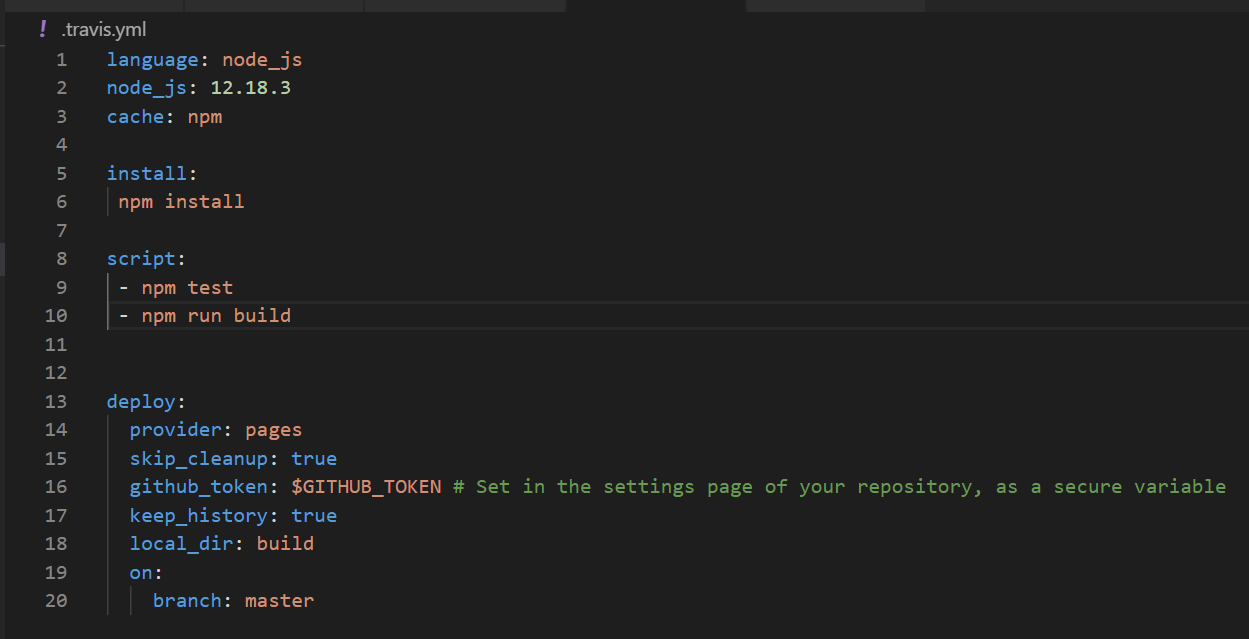
Ce test, il serra a testé si ton « buton » qui a le ID ‘button’ son contexte du buton égale à "Click"

6- créer un compte de Travis Ci sur le site <https://travis-ci.org/>, lié ton compte avec github



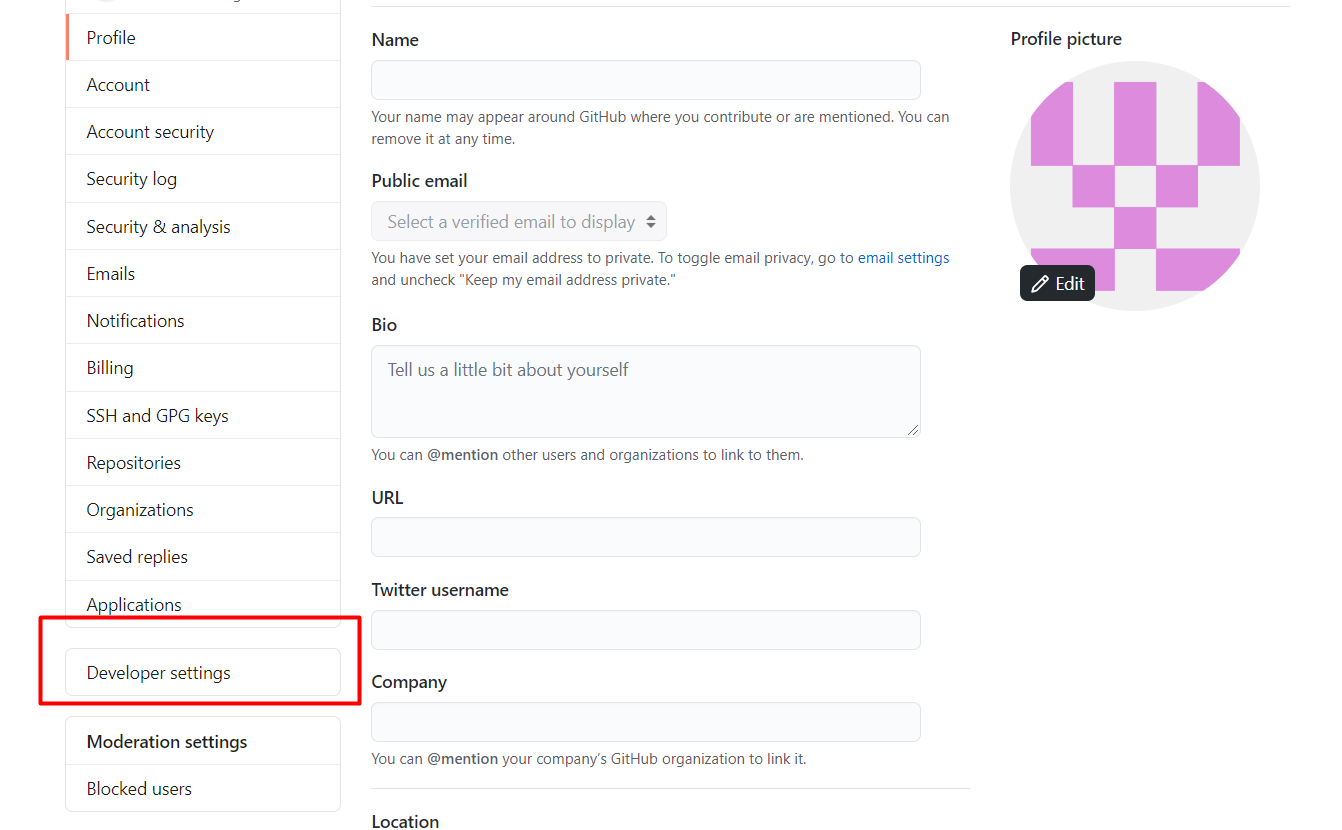
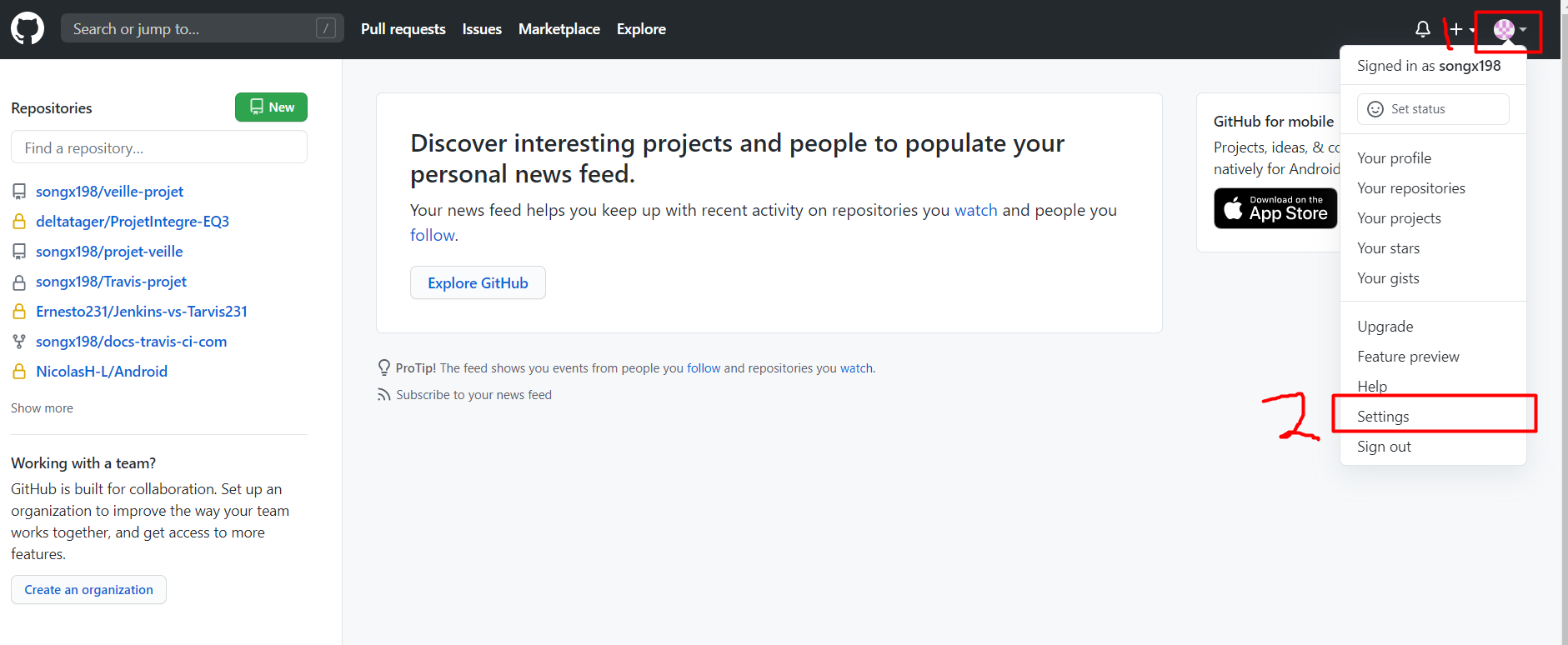
7- ajouter un fichier « .travis.yml » dans ton projet de REACT, il doit être à la même position que ton « src »

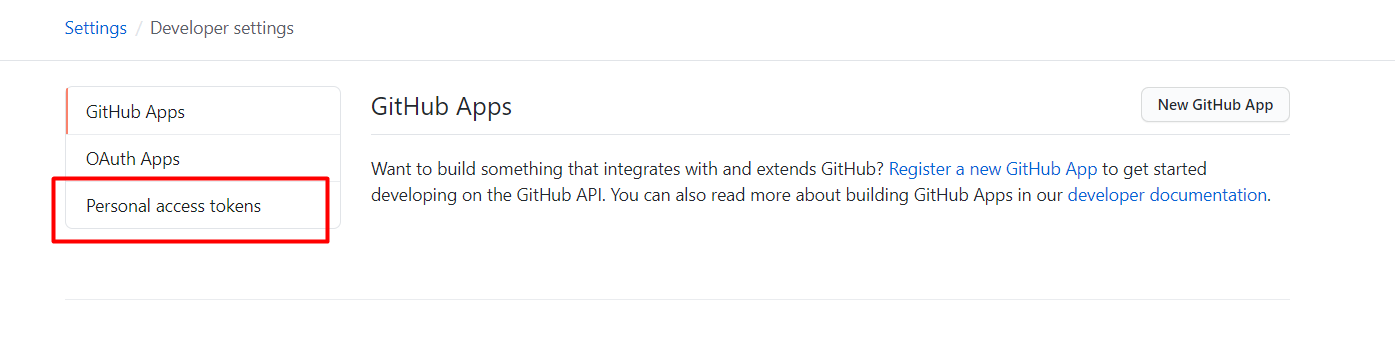
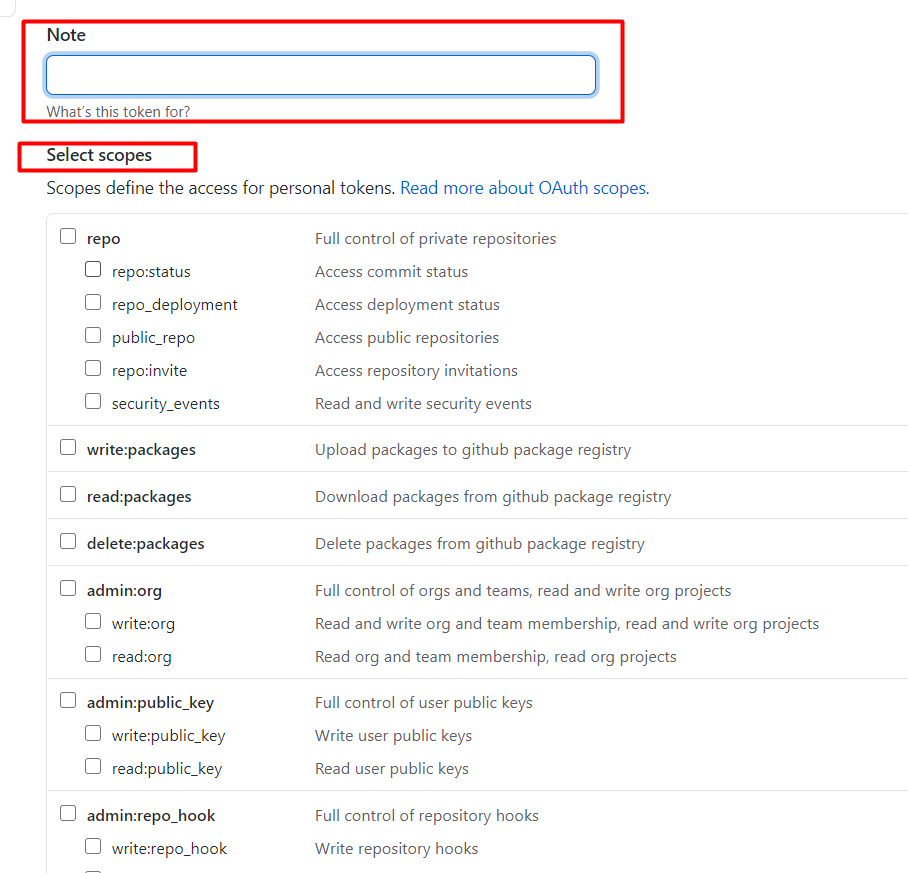
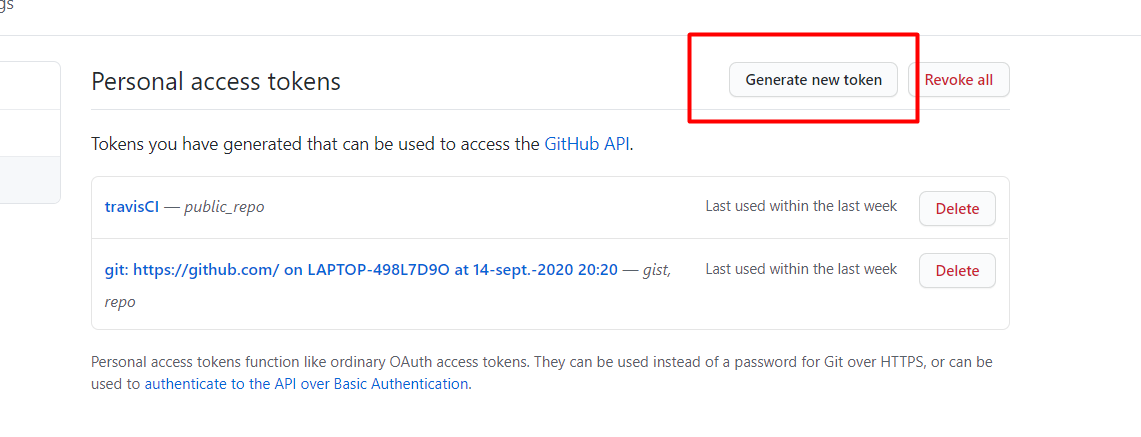
8- ajouter dans le fichier « .travis.yml » les paramètres que tu veux utiliser (on peut suivre le site sur travis-ci)

Dans notre exp : 

Important d’écrire le « npm test » il serre à run notre tests

github\_token se trouve dans :

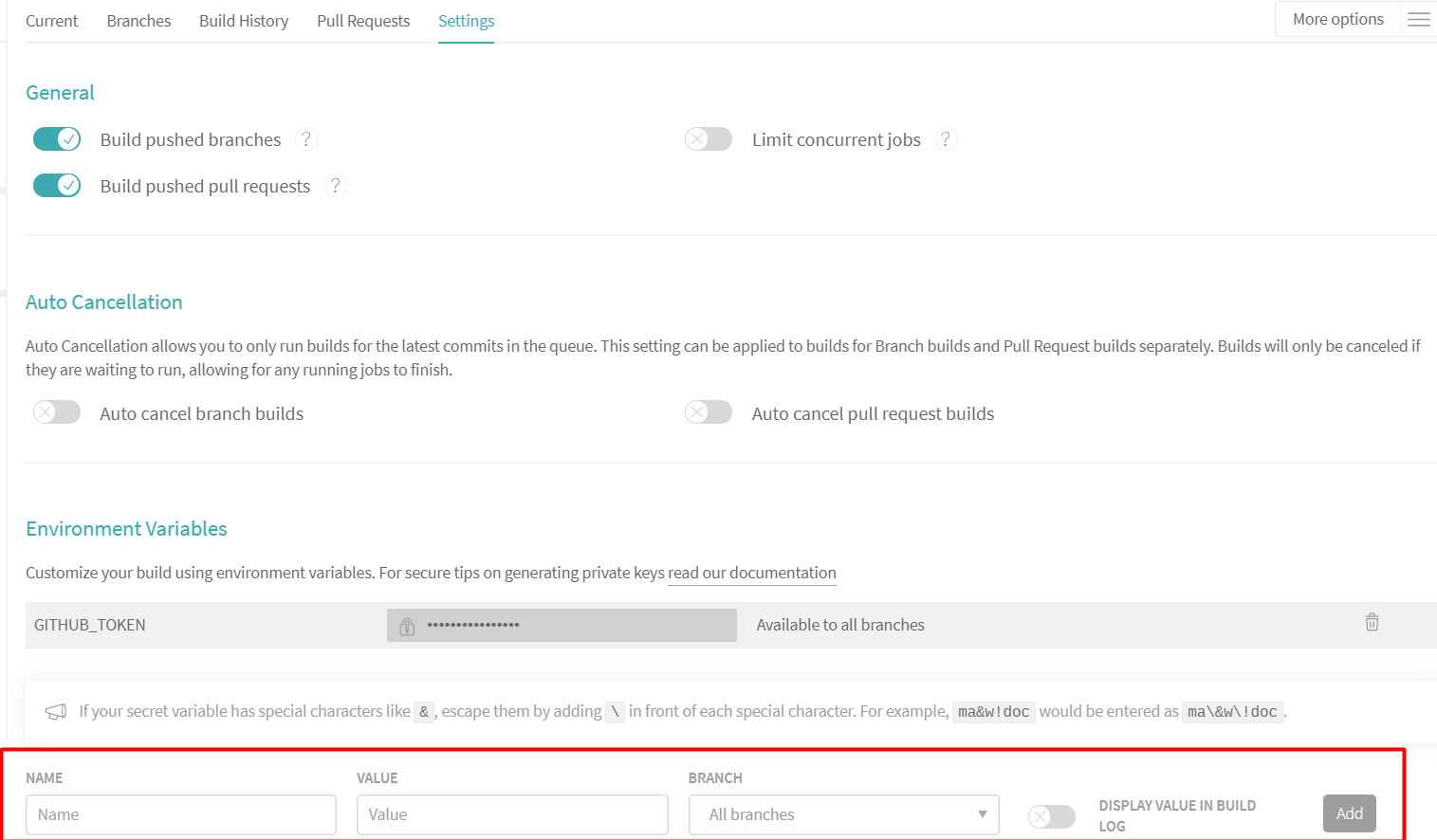
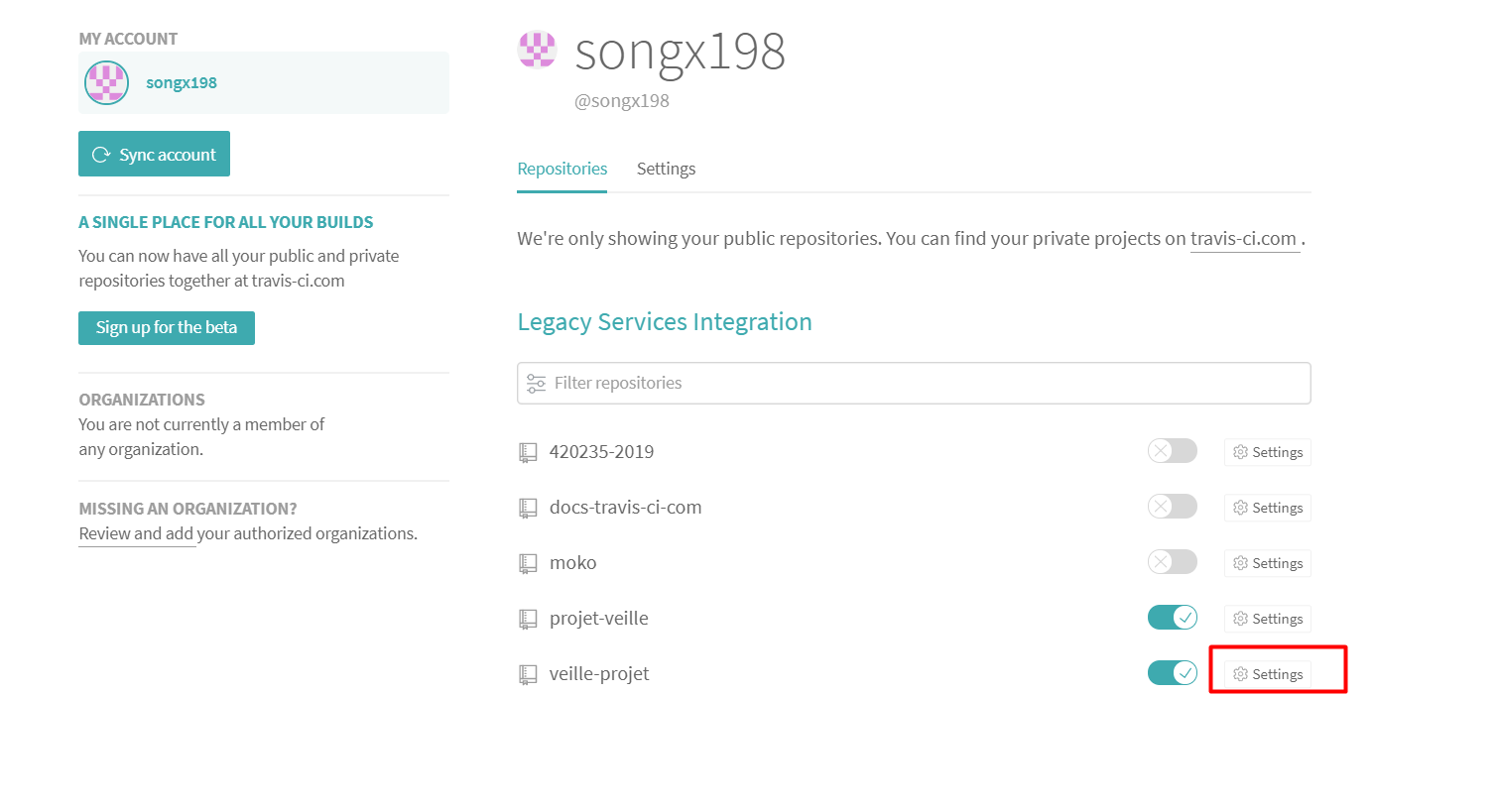


9-mettre le github\_token dans le repository de ton travis-ci :

le nom : GITHUB\_TOKEN

le value : le token de GITHUB d’étape 8



10- fait un commit et push dans le GITBASH

(git add .

git commit -m "le message"

git push origin master)

11- maintenant tu peux aller voir dans le paramètre de ton Travis-CI pour regarder si les tests sont-t-y tous passé ou non. 