La municipalité de SAINT-PAUL DU TERRAY, située en zone d'urbanisation nouvelle a vu s'accroître rapidement sa population durant les cinq dernières années. Elle vient de franchir le cap des 5 000 habitants.

Pour mieux répondre aux besoins importants de la nouvelle population, le conseil municipal a décidé de développer l'information des services municipaux, notamment le service Développement de la vie locale (DVL).

Le service DVL a pour mission l'action culturelle et sportive et les relations avec les nombreuses associations de la commune.

En effet, le centre de loisirs de la commune organise chaque année des séjours extra-scolaires, comme les classes de découverte par exemple, pour les enfants des écoles primaires. Le service DVL est chargé de l’information aux familles mais aussi du calcul et du recouvrement de la participation financière des familles.

Actuellement le site affiche la liste des séjours prévus ou à venir pour l’année scolaire en cours. Une capture d’écran est disponible en Annexe 1.

**Le service informatique a développé ce logiciel il y a plusieurs années et souhaite faire du refactoring pour utiliser les normes et les bonnes pratiques actuelles.**

*Ce projet s’inspire du travail de B Pesquet et vous pourrez vous aider du site Openclassrooms : https://openclassrooms.com/courses/evoluez-vers-une-architecture-php-professionnelle*

TRAVAIL A FAIRE

# Itération 1 : Recréer un environnement de développement pour exécuter le site actuel.

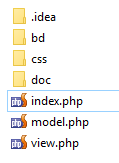
On utilisera un framework de développement : Eclipse ou PHPStorm. Récupérer une licence pour PHPStorm si besoin.

Initialiser la BD : crebd.sql + sejour.sql

Exécuter le logiciel

Lire index.php. Quels sont les défauts de ce code ?

# Itération 2 : réécrire le code en appliquant le principe de Responsabilité unique

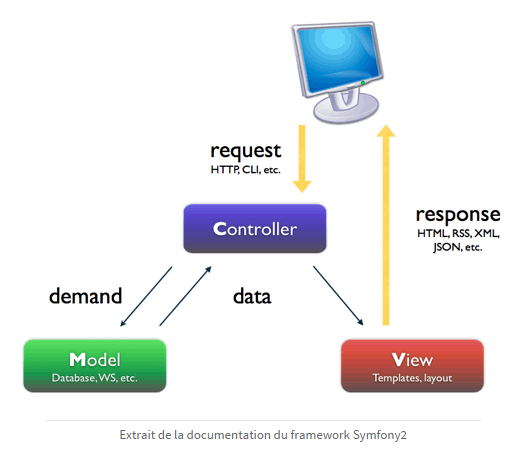


On utilise un dépôt Git pour pouvoir écrire plusieurs versions de cette application. Créer un compte. Créer une organisation. Créer un dépôt.

On veut aboutir à 3 fichiers distincts comme sur l’arborescence ci-contre :

Tagger cette version sur git avec le tag V1.2

Commenter le schéma suivant et chacun des composants de l’architecture MVC. :



# Itération 3 : intégrer le framework SILEX

[Cf SILEX.docx](Silex.docx)

Tagger cette version sur git avec le tag V1.3

# Itération 4 : modélisation objet de l’accès aux données

Nous allons restructurer cette partie en introduisant une modélisation orientée objet des données métier.

1. Ecrire la classe SEJOUR correspondant à la table (avec constructeur et accesseurs).

<?php

namespace stpaul\Domain;

class Sejour

{

/\*\* TODO \*/

}

On utilise l’espace de nom stpaul\Domain. Le fichier Sejour.php doit donc se trouver dans le dossier src\ stpaul\Domain.

1. On va remplacer PDO par Doctrine DBAL ( (DataBase Abstraction Layer).

Quel est l’intérêt de DBAL ?

Ajouter Doctrine/DBAL à COMPOSER.

1. Ecrire le code d’accès aux données en vous inspirant de ce [lien](https://openclassrooms.com/courses/evoluez-vers-une-architecture-php-professionnelle/iteration-4-modelisation-objet-de-l-acces-aux-donnees#/id/r-2614739)
2. Modifier le contrôleur en vous inspirant de ce [lien](https://openclassrooms.com/courses/evoluez-vers-une-architecture-php-professionnelle/iteration-4-modelisation-objet-de-l-acces-aux-donnees#/id/r-2614908)
3. Tagger cette version sur git avec le tag V1.4

# Itération 5 : utiliser le moteur de template twig pour la présentation.

1. Modifier votre application pour intégrer twig en suivant ce [lien](https://openclassrooms.com/courses/evoluez-vers-une-architecture-php-professionnelle/iteration-5-integration-du-moteur-de-templates-twig#/id/r-2617708).
2. Tagger cette version sur git avec le tag V1.5

# Itération 6 : Tests unitaires.

Les élus de la commune demandent une évolution sur la page :

* Affichage de la date de fin de séjour
* formatage de la date en français

Réaliser cette modification.

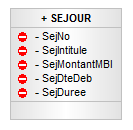
Réaliser les tests unitaires correspondants.

Tagger cette version sur git avec le tag V2.0. Cette version sera mise en production.

# Annexe 1 - Page web du site actuel

****

# Annexe 2 - Extrait de la base de données

SEJOUR

Contient les séjours déjà réalisés ou ceux prévus

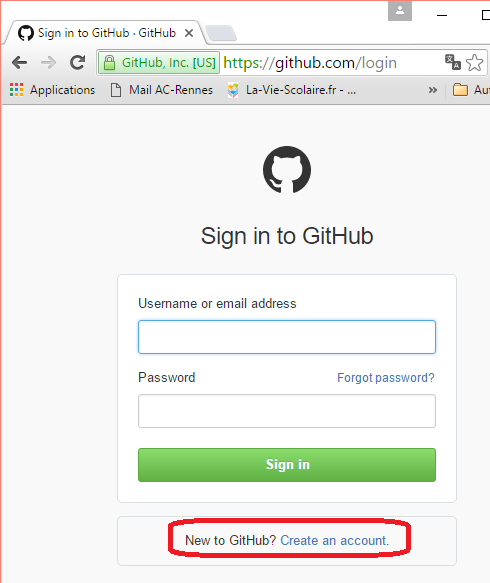
Exemple d’Intitulé de séjour : Classe de découverte

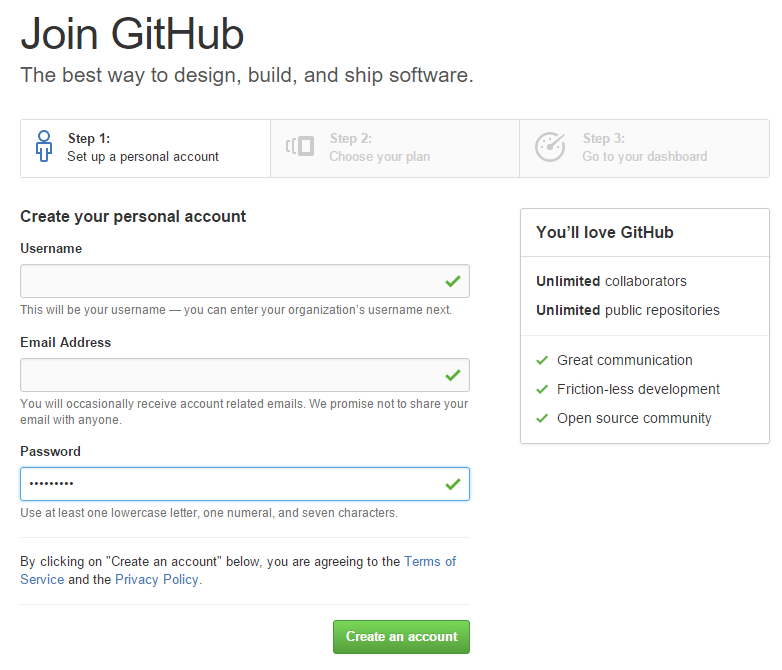
SejMontantBI : montant brut individuel d’un séjour

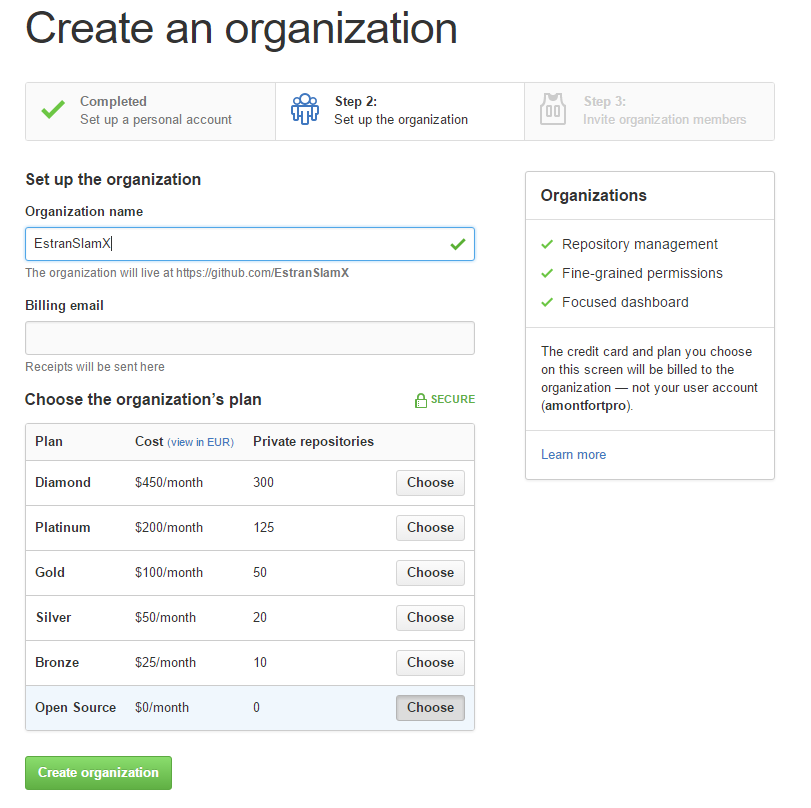
Duree : durée exprimée en nombre de jours

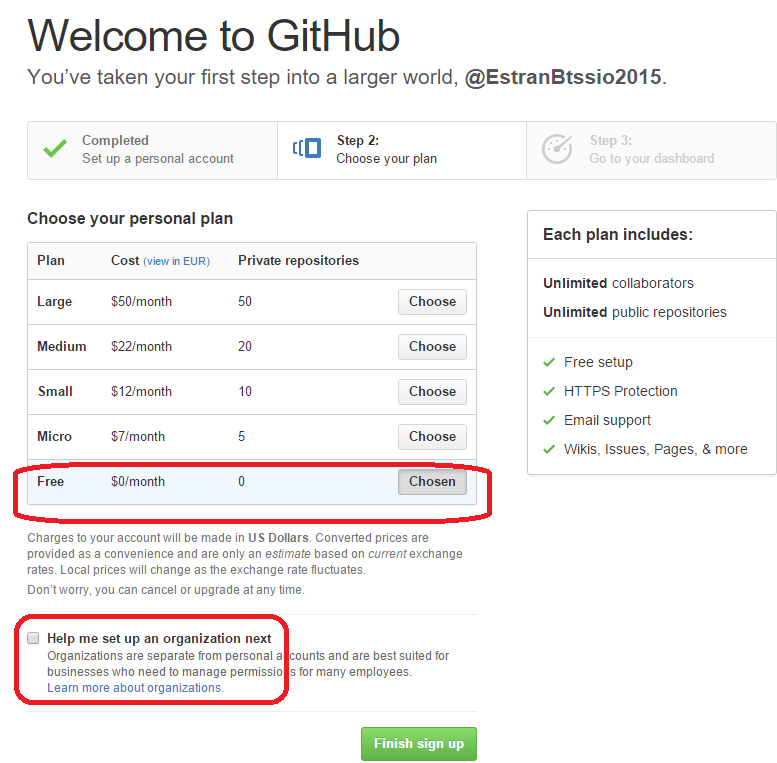
# Annexe 3 - Créer un dépôt sur Github

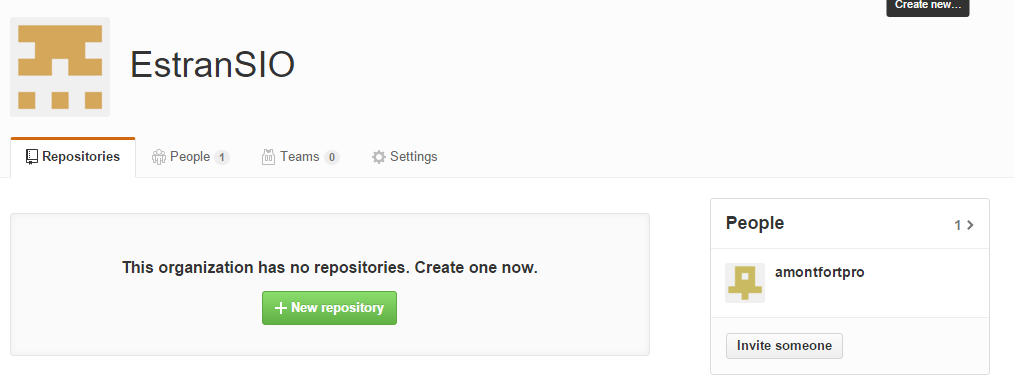
[**https://git-scm.com/book/fr/v1/D%C3%A9marrage-rapide**](https://git-scm.com/book/fr/v1/D%C3%A9marrage-rapide)

****

****

****

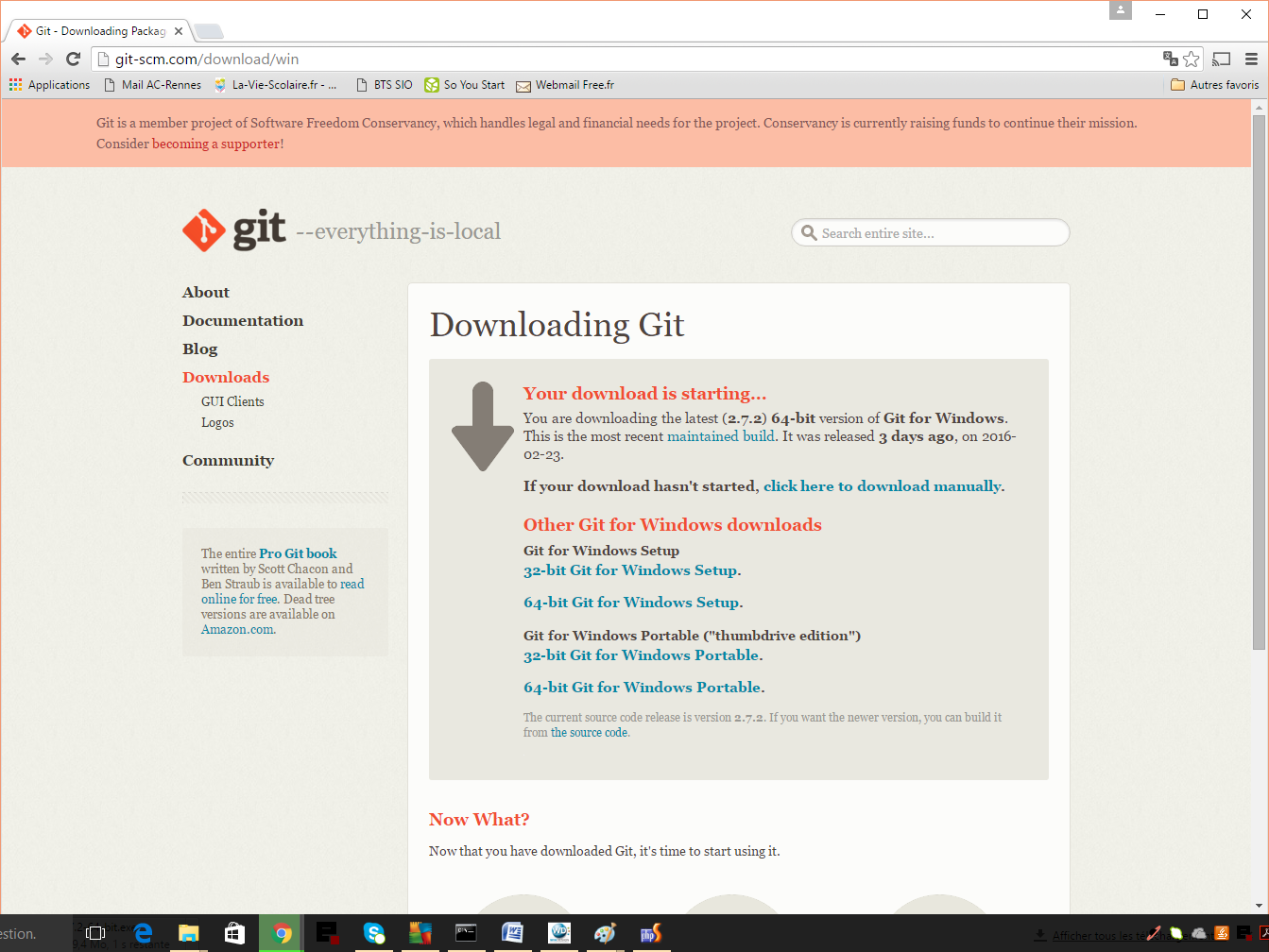
****

****

# Annexe 3 - Configurer PHPStorm pour gérer les versions avec GIT et créer un dépôt local

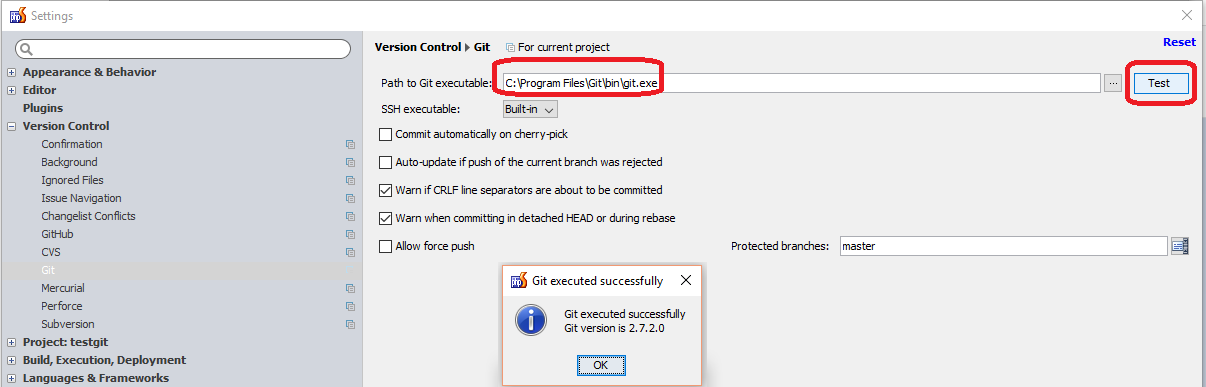
Installer git sur votre poste

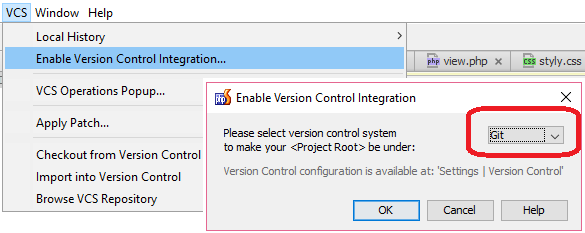
* A partir des ressources pédagogiques
* A partir de l’URL suivante :

****

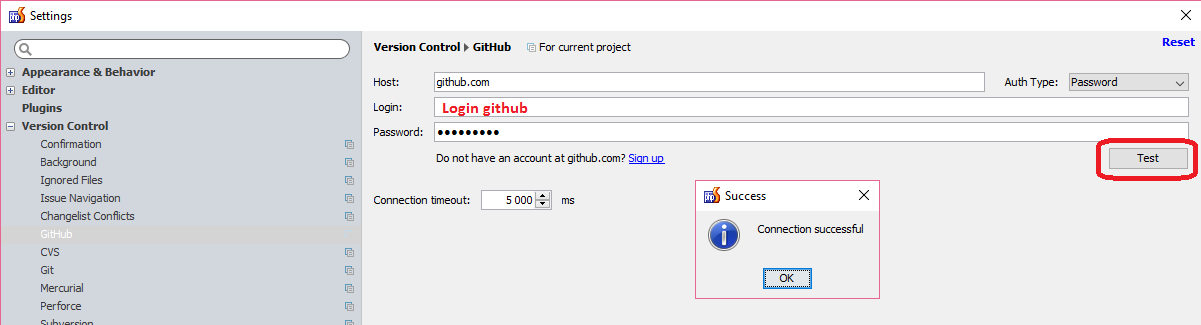
Configurer votre projet dans le menu Settings

**Activer le contrôle de version avec GIT.**

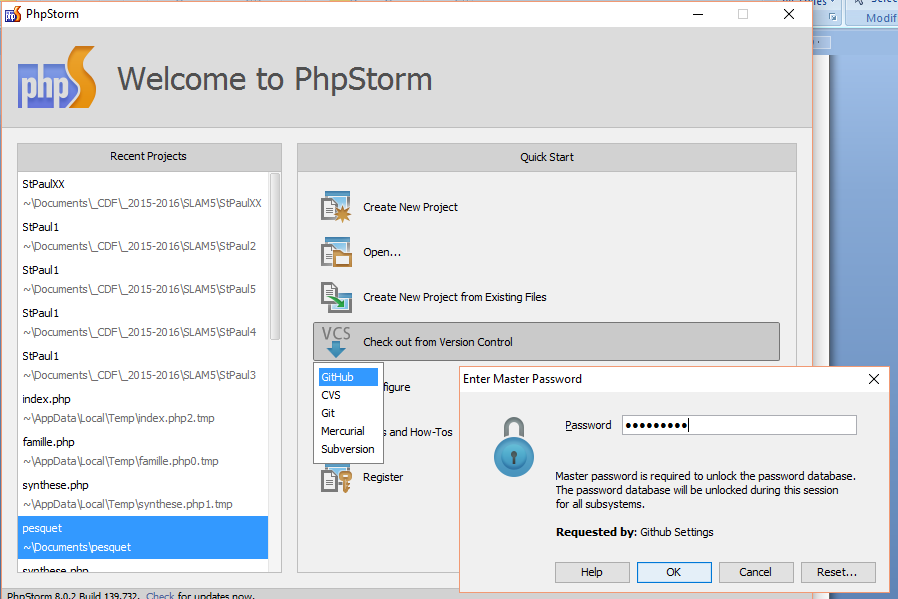
****

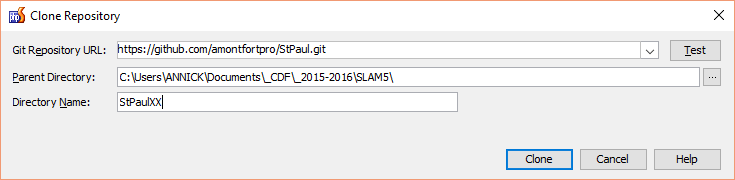
****

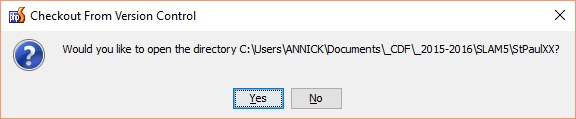
**Indiquer votre dépôt distant sur github**

****

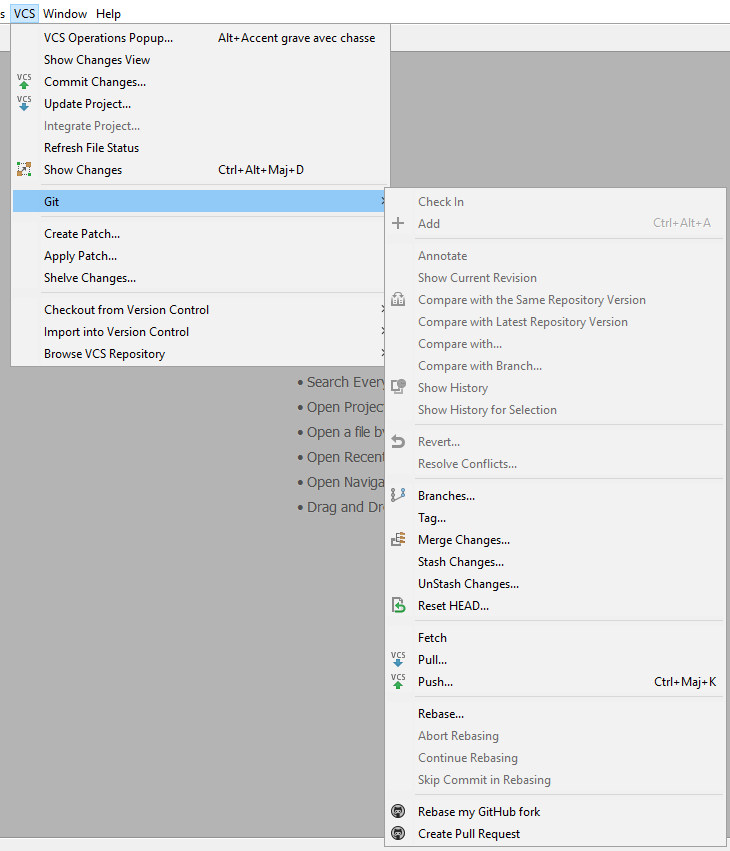
Créer un nouveau projet

****

****

****

**Vous disposez alors des fonctions de gestion de versions de git :**

****

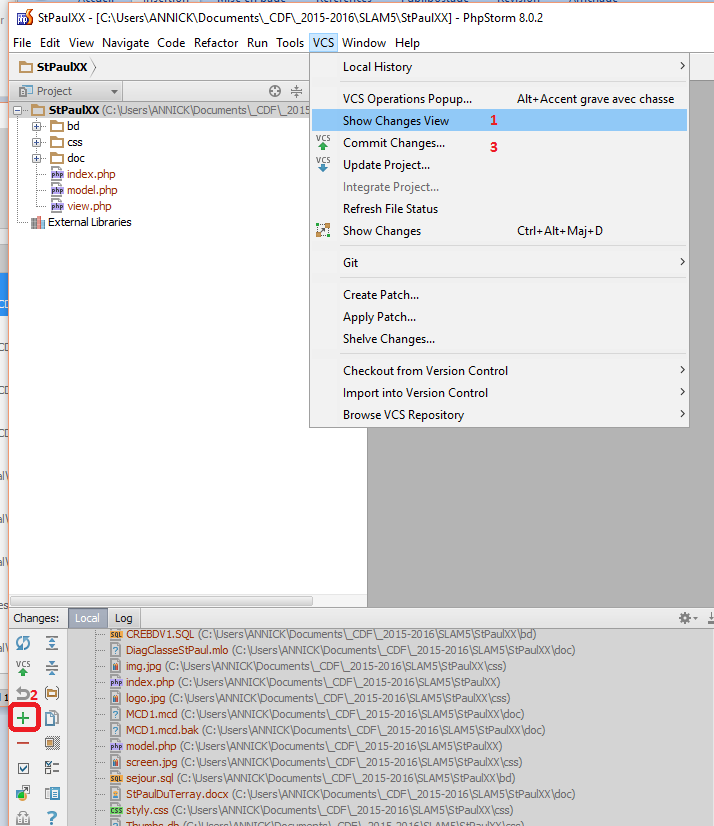
Recopier dans StPaulXX vos sources (sauf le dossier .idea) et synchroniser PHPStorm si besoin.

Mettre à jour le serveur distant

Pour remonter les fichiers sources sur le serveur distant github, il faut :

* Créer une changelist pour indiquer que tous les fichiers ont été modifiés et forcer ainsi le 1er commit.
* Exécuter le Commit
* Exécuter un Push  pour remonter les fichiers.

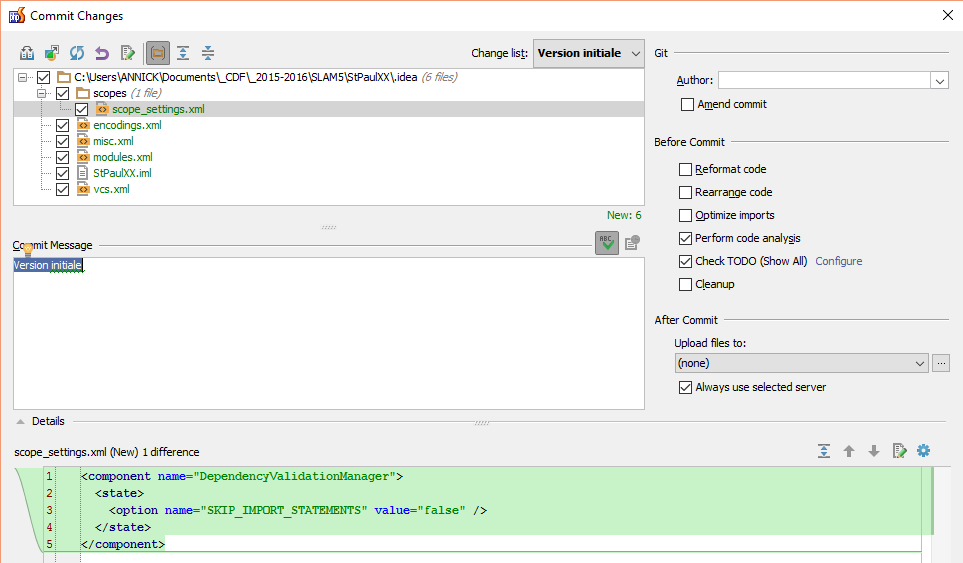
Créer la Changelist :

****

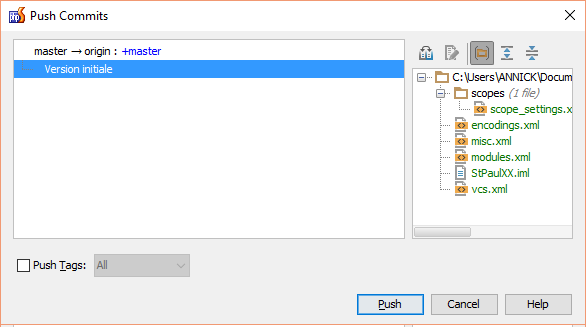
Il faut alors donner

* un nom à la liste : Version initiale par exemple
* s’authentifier.

Faire le Commit

****

Faire le Push

****