TUTO GITHUB

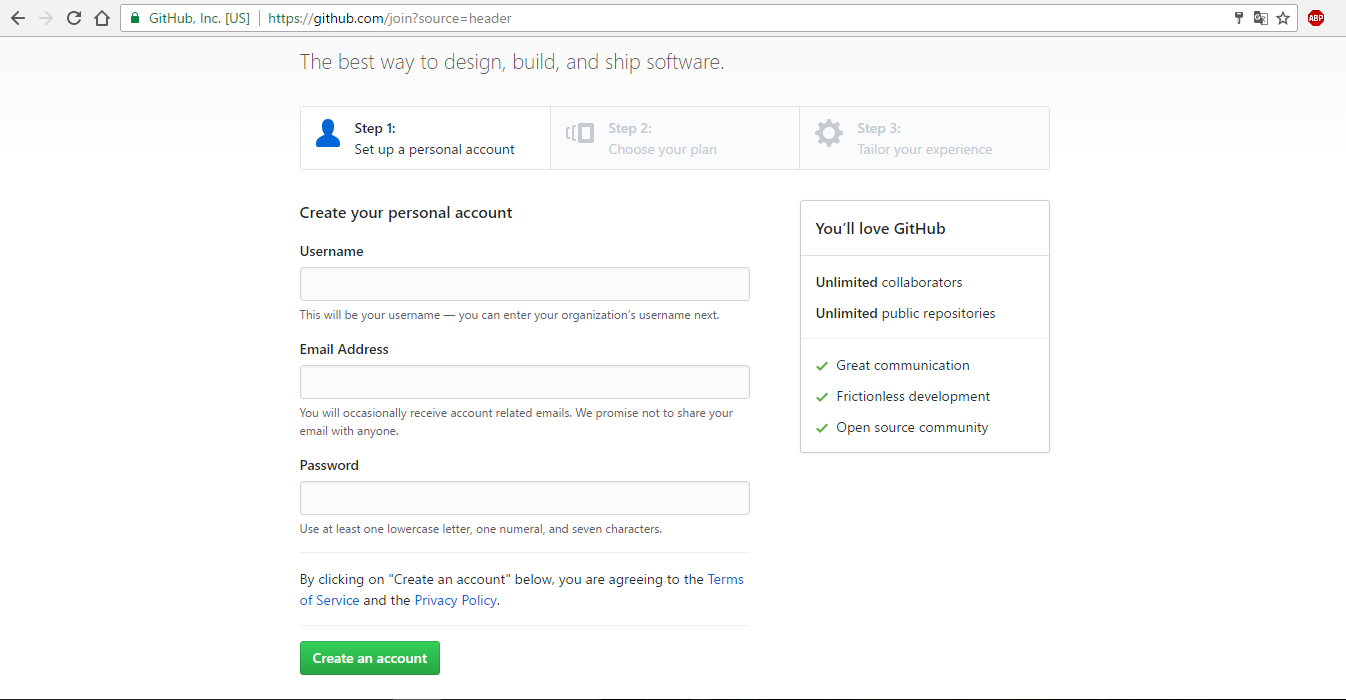
Voici un tuto Github pour créer un repertoire de dépot pour les projets collaboratifs

Tout d’abord se rendre sur l’url : <https://github.com/>

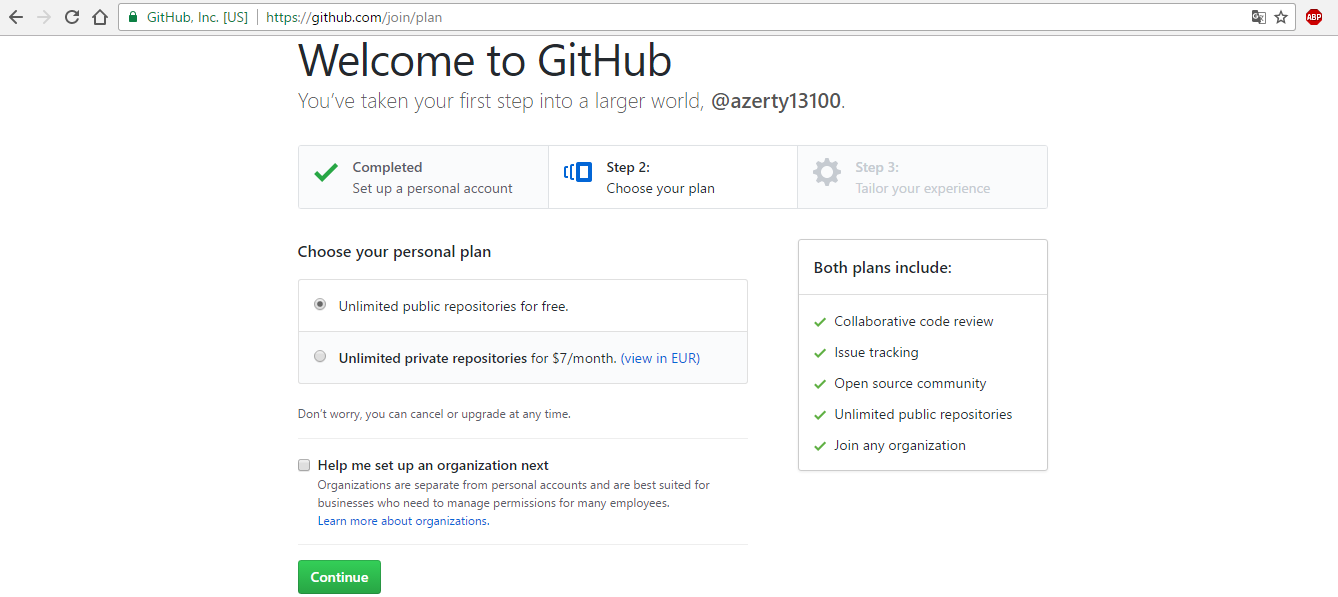
SI vous avez déjà un compte Git vous pouvez sauter l’étape 1

Etape 1 : Inscription

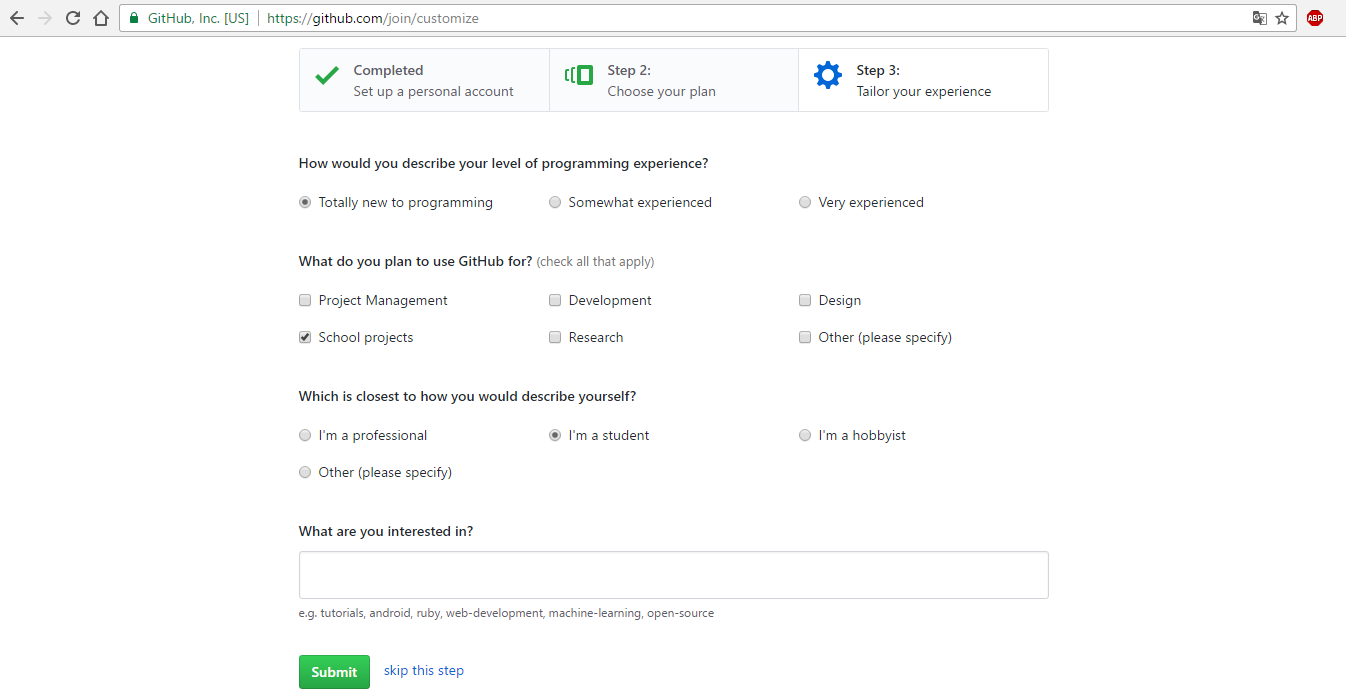
Pour créer un compte aller sur Signup en haut de page, puis cet interface apparait ->



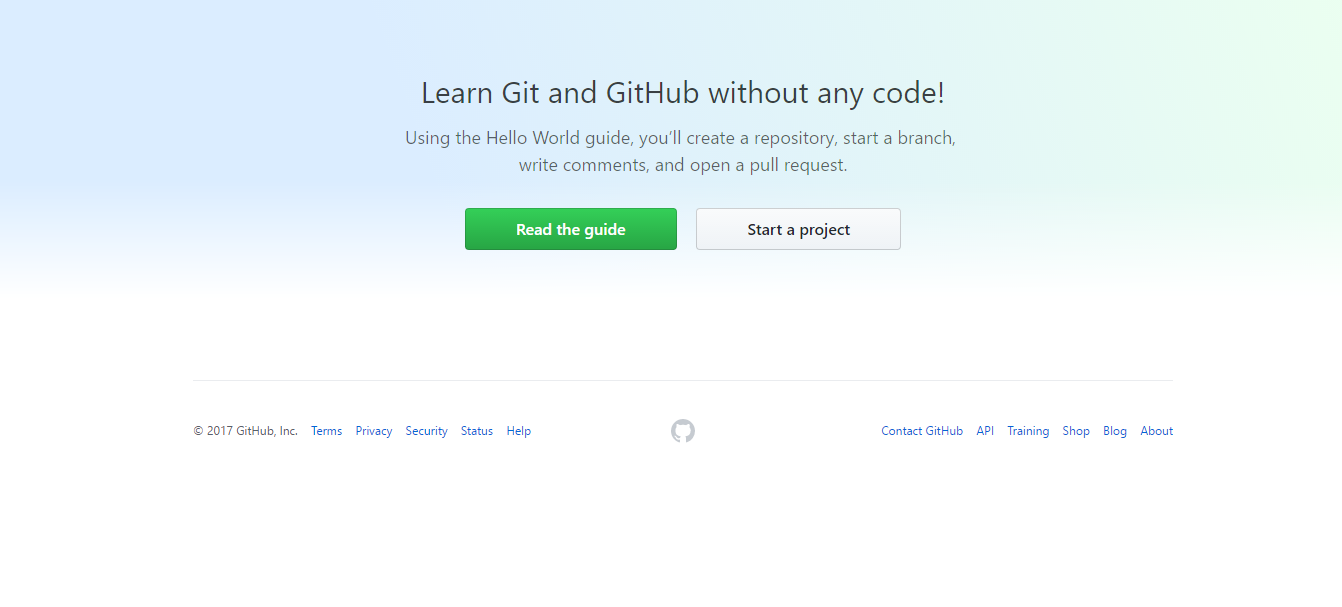
Une fois les champs remplis cliquer sur « Create an account »



Choisir la version gratuite « unlimited repositories for free » et « continue »



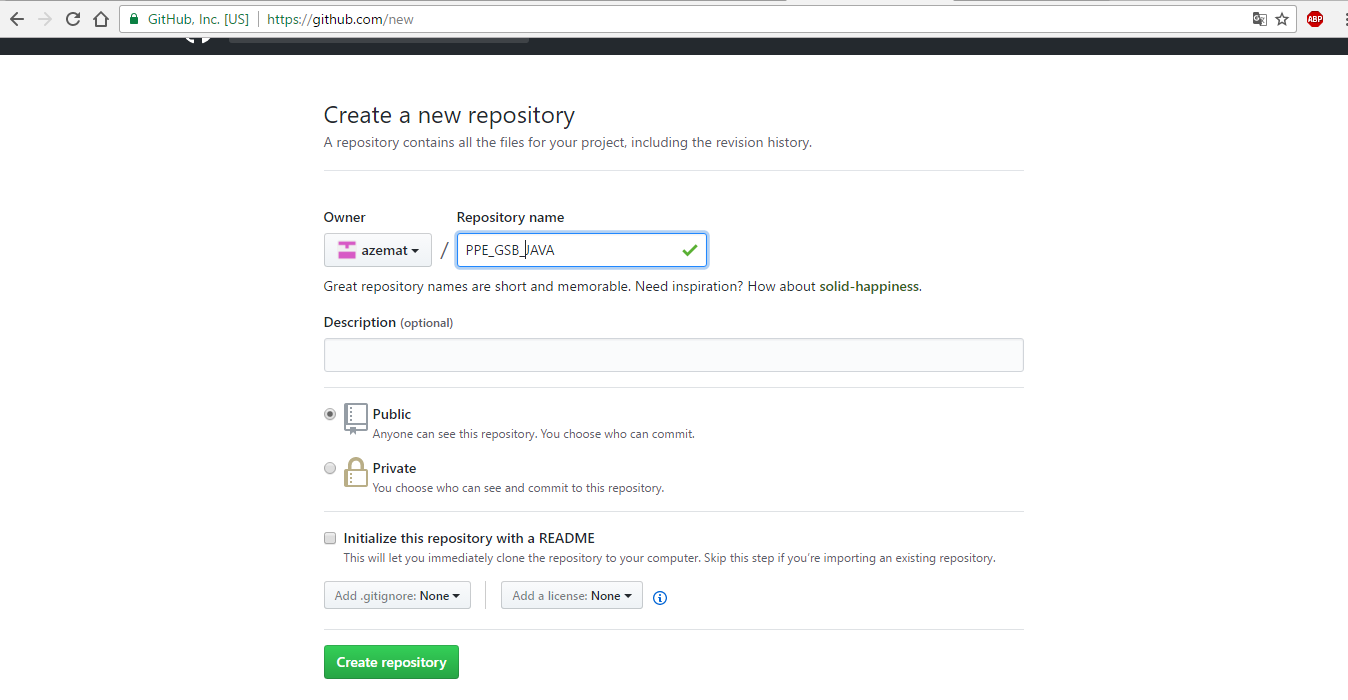
Derniere étape pour la création de compte Git. Sélectionner les champs qui correspondent à notre situation puis cliquer sur « Submit »



Sélectionner « Start a project »

Puis il est nécessaire de valider notre adresse email avant de pouvoir continuer.

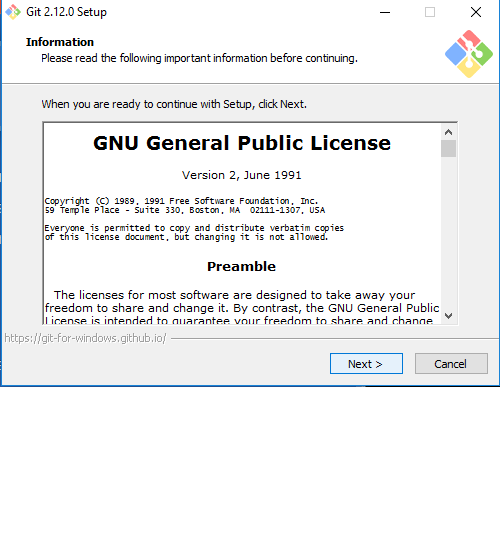
Etape2 : Création repository+ téléchargement Git



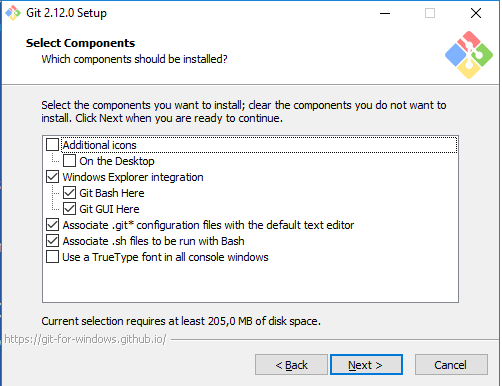
Voilà le repository est créé, il faut maintenant télécharger GIT sur

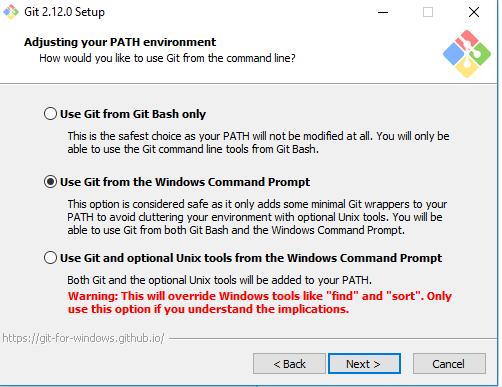
<https://git-for-windows.github.io/>

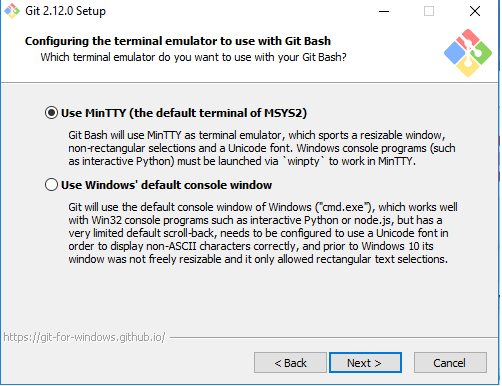
Une fois le téléchargement terminé cliquer sur ouvrir et suivre l’installation

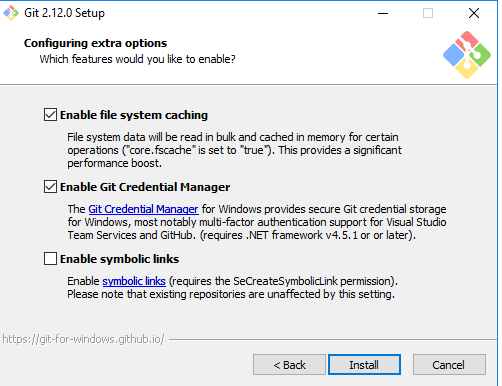


Continuer l’installation





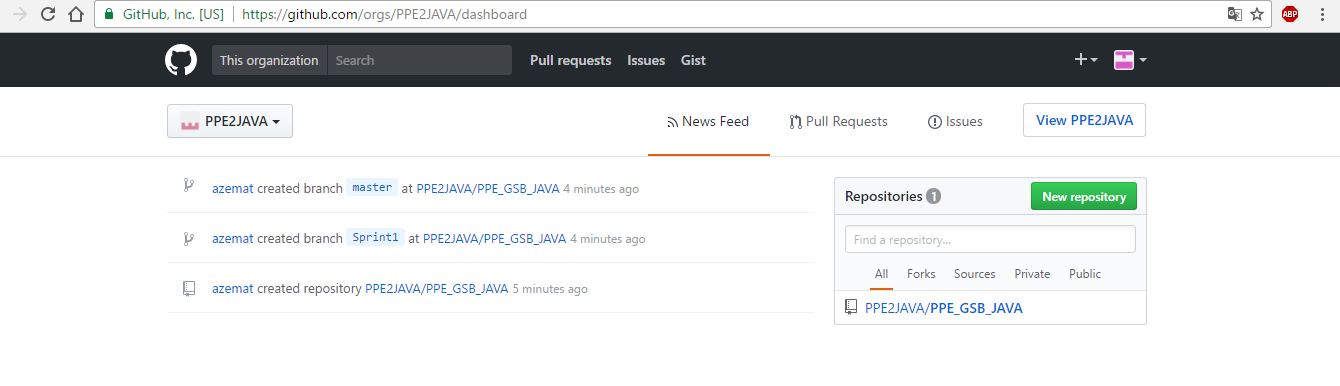




Voilà l’installation de git sur le pc est terminé

Sur Github.com nous avons créé une organisation avec les 4 membres de l’équipe. Ansi qu’un repository

Nous avons créé une nommé Sprint1, c’est la que seront deposé les modifications de code, cela permet de laisser la branche Master intacte et ainsi de voir les modifications apportées et/ ou de faire un retour en arriere si besoin ajouter un rapport de visite



Voici les commandes Git les plus importantes

Ouvrir invité de commandes

Commandes :

Git init

git add + le nom du fichier : Pour ajouter un fichier à l’index

git commit :de l'index au repository(répertoire)

git reset :git checkout : revenir à une version précédente

git diff : sert à afficher les modifications apportées entre les 3 arbres

git log : sert à afficher l'historique

git status : ça donne l'état de chaque fichier

git branch : permet de créer, lister et supprimer des branches

git checkout : permet de déplacer la référence head vers une nouvelle branche

git checkout –b : permet de faire git branch + git checkout.