Dans un 1er temps, nous allons développer en angular 7 la partie front de ce site intitulé myCms présentant des articles que la personne gestionnaire de ce site pourra éditer à loisir.

Puis dans un second temps, après avoir conçu la partie visible de l’iceberg aux internautes, nous développerons la partie back-end de ce site. Nous exposerons les services implémentés avec SpringBoot ( [https://codelabs.developers.google.com/codelabs/cloud-springboot-cloudshell/index.html?index=..%2F..index#0](https://codelabs.developers.google.com/codelabs/cloud-springboot-cloudshell/index.html?index=..%2F..index" \l "0) ) dont a besoin l’IHM Angular et qui restitue les données stockées dans Google App Store ou Firebase.

Logiciels à installer :

- sur chromebook :

- visual studio code : <https://chromeos-cookbooks.firebaseapp.com/web.html>

<https://www.youtube.com/watch?v=WZvEbMUMf7s>

- Node : <https://chromeos-cookbooks.firebaseapp.com/web.html>

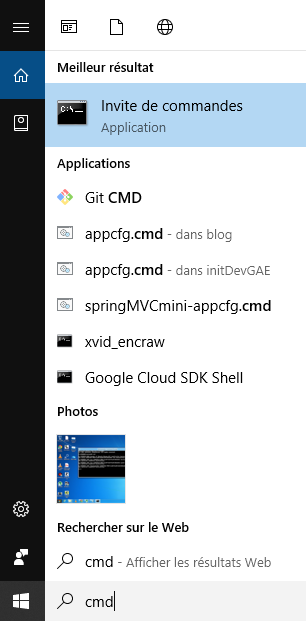
- Angular CLI : <https://chromeos-cookbooks.firebaseapp.com/web.html>

-installer git si ce n’est pas déjà fait.

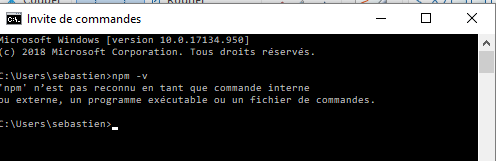
<https://chromebook.home.blog/2019/03/18/installing-java-and-eclipse-ide-on-a-chromebook/>

- sur windows 10 :

Via l’invite de commande(en l’invoquant en tapant cmd+[enter]), nous allons vérifier si le package manager de Node.js (NPM) est déjà installé sur notre système (Windows 10) :

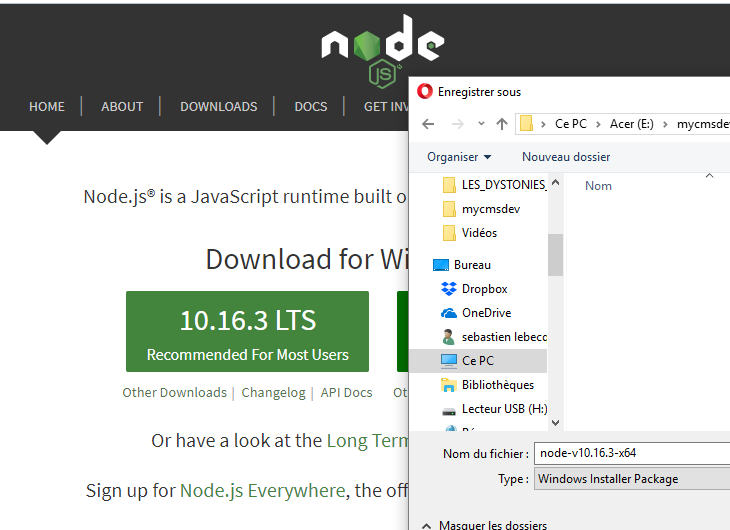


Pour ce faire (après avoir invoqué l’invite de commande), invoquons la commande npm -v afin de savoir quelle est la version de npm installée :



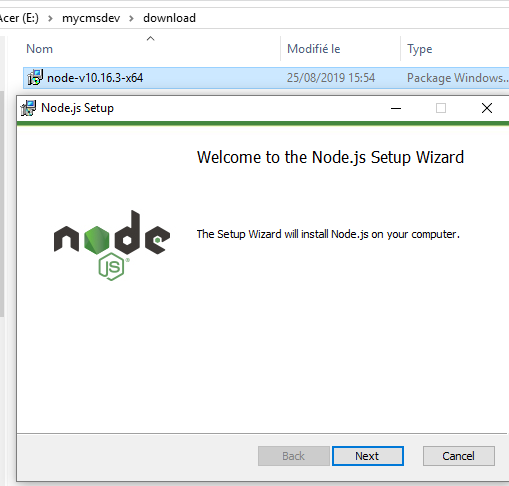
Comme npm n’est pas installée sur le système, rendons-nous sur le site de node.js :

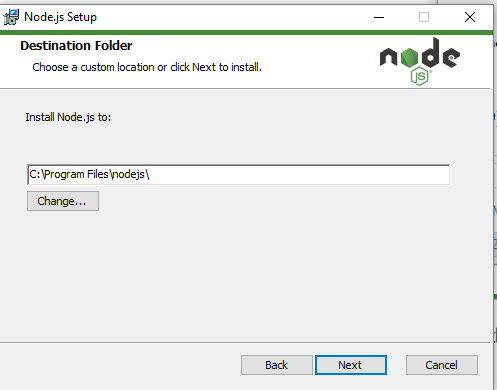
Et de là, téléchargeons le MSI rendant l’installation effective de Node.js donc Npm (y étant inclus) :

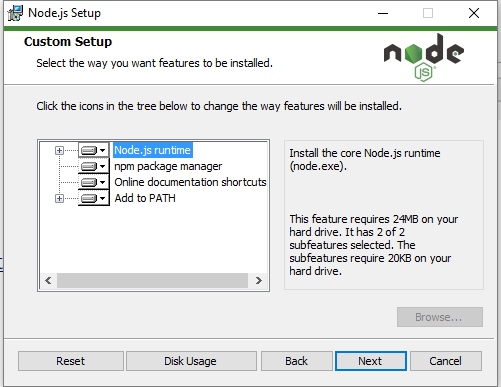


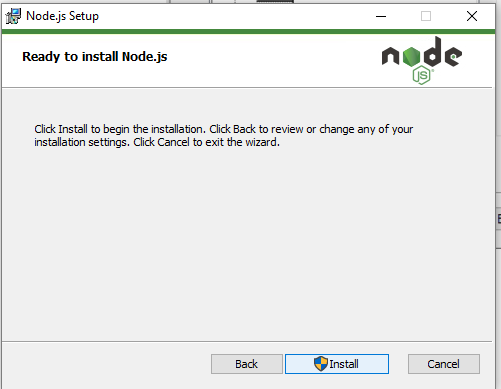
- angular 7 :

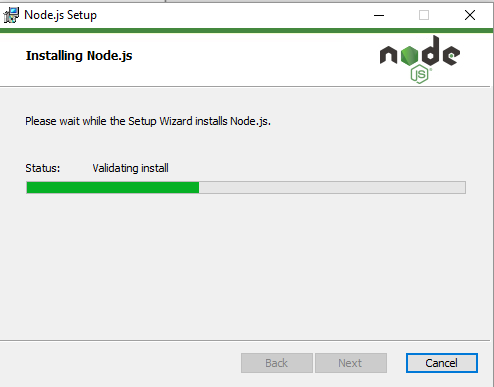
Une fois l’executable chargé, le Wizard de Node.js apparaît permettant de personnaliser l’installation :



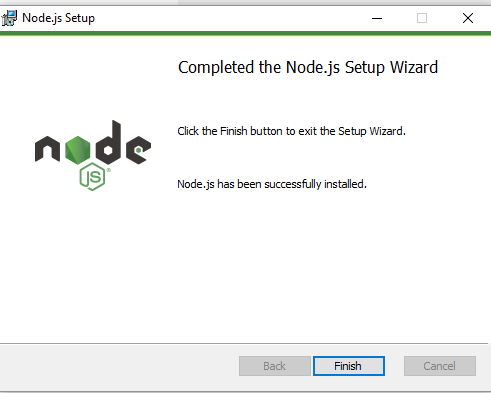




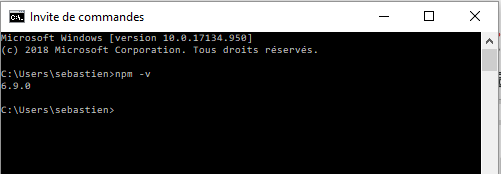
[](https://www.javatpoint.com/angular-7-installation)

[](https://www.javatpoint.com/angular-7-installation)

[J](https://www.javatpoint.com/angular-7-installation)usqu’à la fenêtre finale confirmant l’installation de Node.js :

[](https://www.javatpoint.com/angular-7-installation)

En effet, l’entrée de la commande « npm -v » retourne la version de npm installée avec node.js

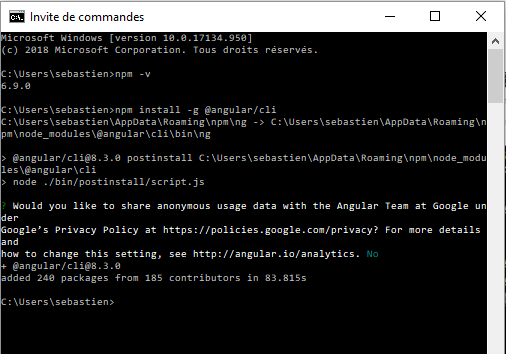


A présent, nous pouvons procéder à l’installation d’Angular CLI :

(<https://www.javatpoint.com/angular-7-installation> )

en entrant dans l’invite de commande :

npm install -g @angular/cli



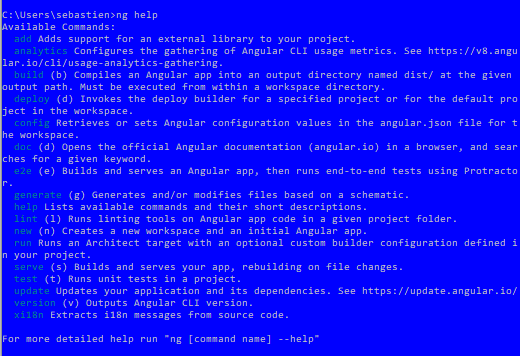
Pour vérifier qu’Angular CLI est bien installé avec une méthode de vérification à la ligne de commande, entrons :

ng -version



A présent, pour créer notre premier composant avec Angular, voyons ce que que retourne la commande :

ng help



La commande « new » permet de créer un nouveau workspace et d’initialiser notre application Angular.

Sur mon ordinateur, le répertoire concernant ce projet s’intitule « mycmsdev » .

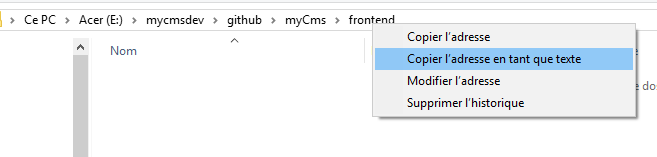
Dans ce répertoire, j’ai pu auparavant créer un répertoire « download » où j’ai téléchargé depuis l’internet le fichier msi d’installation de Node.js. (d’autres téléchargements y viendront par la suite)

Je crée un autre répertoire « github » dans lequel je synchroniserai mon projet « myCms » dans un 1er temps avec uniquement sa partie front-end puis sa partie back-end ultérieurement.

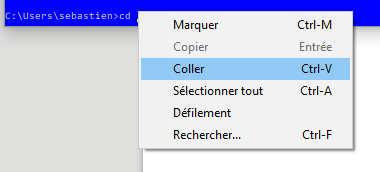


Pour l’instant, je ne m’occupe que de la partie frontend.

Alors, plaçons-nous dans ce répertoire et créons-y notre premier composant:



Entrons d’abord la commande cd (pour Change Directory) dans l’invite de commande et collons-y le chemin dans le File System préalablement copié :







Dans ce répertoire, créons notre premier composant que l’on nommera article (car mon propre CMS s’occupe de gérer des articles), donc tapons :

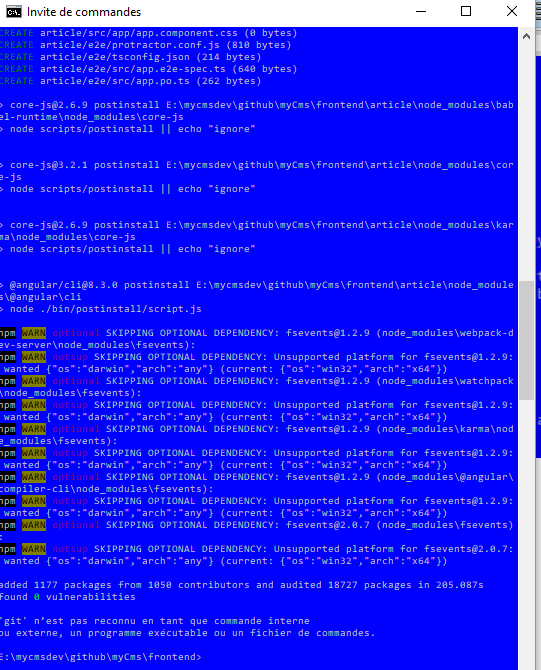
ng new article

A la question si l’on désire utiliser le mécanisme de routage des applications Angular, la réponse est Y

A la question pour savoir que feuilles de style nous utilisons, choisissons le format par défaut en tapant sur le bouton entrée :

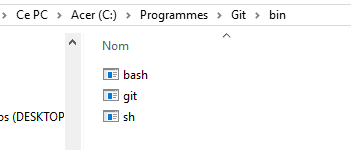


Au bout de quelques dizaines de secondes :

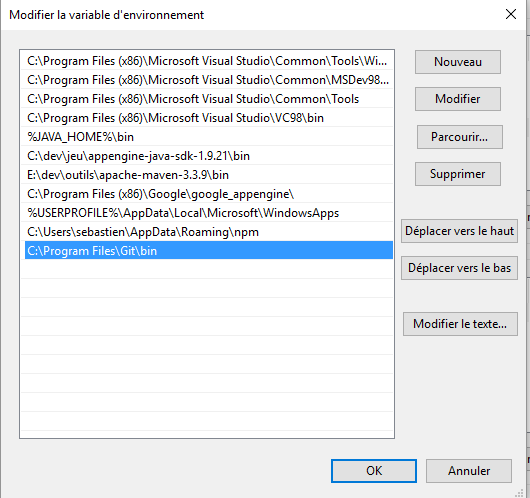


Il semble que le projet a été créé, mais git n ‘était pas reconnu sur mon système, pourtant, par le passé, j’ai déjà utilisé Git. En effet, je le retrouve dans le liste des programmes de windows 10, je vais dans C:\Prammes\Git\bin

et j’en copie l’emplacement que je vais renseigné dans la variable PATH :

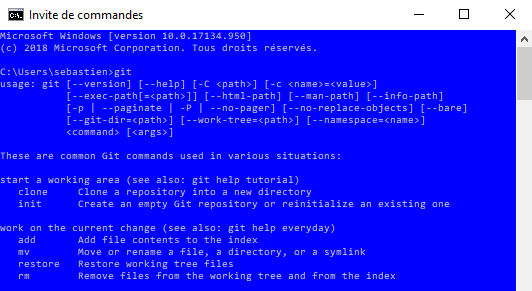


(cf google : variable d’environnement PATH dans Windows 10)

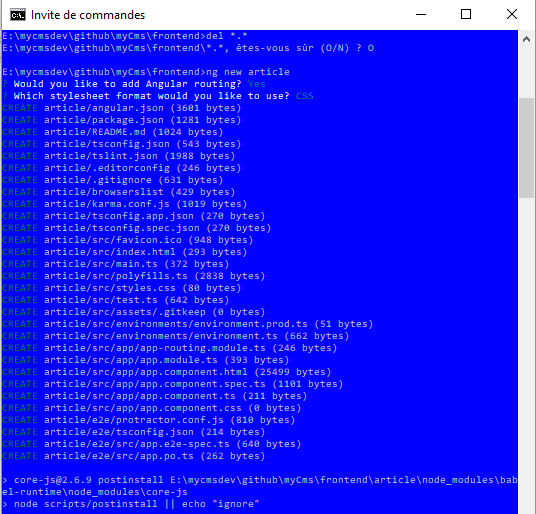


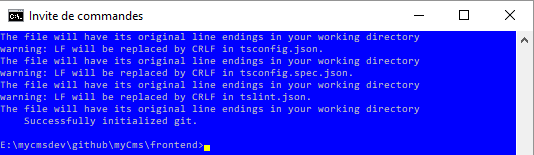
Pour ceux qui n’auraient pas installé git sur leur poste, je les conseille de se rendre sur le site officiel de git en procédant à son téléchargement et à son installation : <https://git-scm.com/downloads>

A présent, la commande git est reconnue :



Par mesure de sécurité et parce que nous sommes à l’étape d’initialisation d’un projet, nous pouvons recommencer le process de création du component sans oublier d’effacer les fichiers:

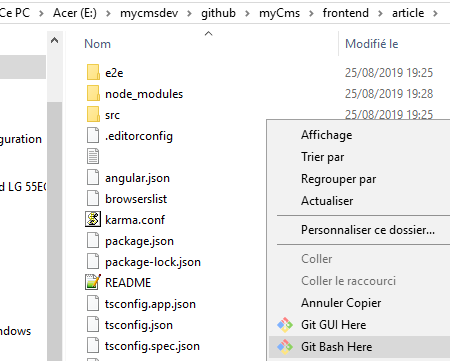




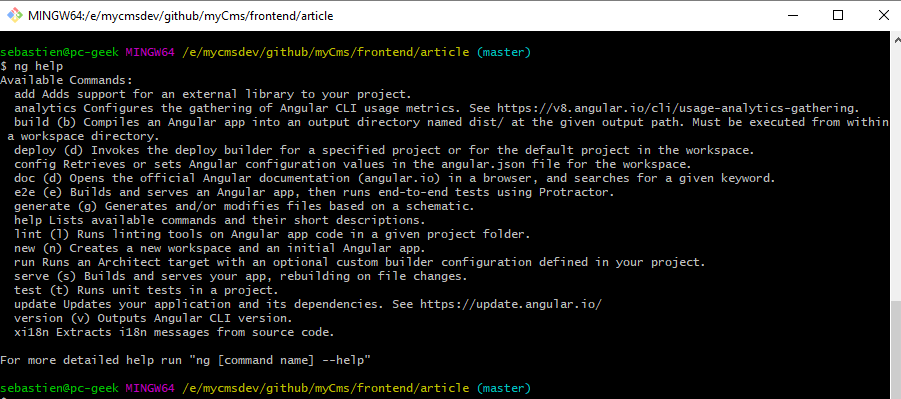
Finalement, le projet est correctement initialisé sous Git.

Vérifions l’execution de ces sources.

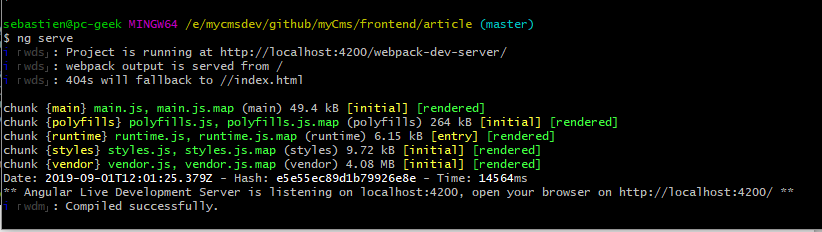
Pour ce faire, voici un autre moyen pour appeler la ligne de commande, via le raccourci ‘Git Bash Here’ disponible dans le menu contextuel en nouspositionnant dans le répertoire de notre application Angular :



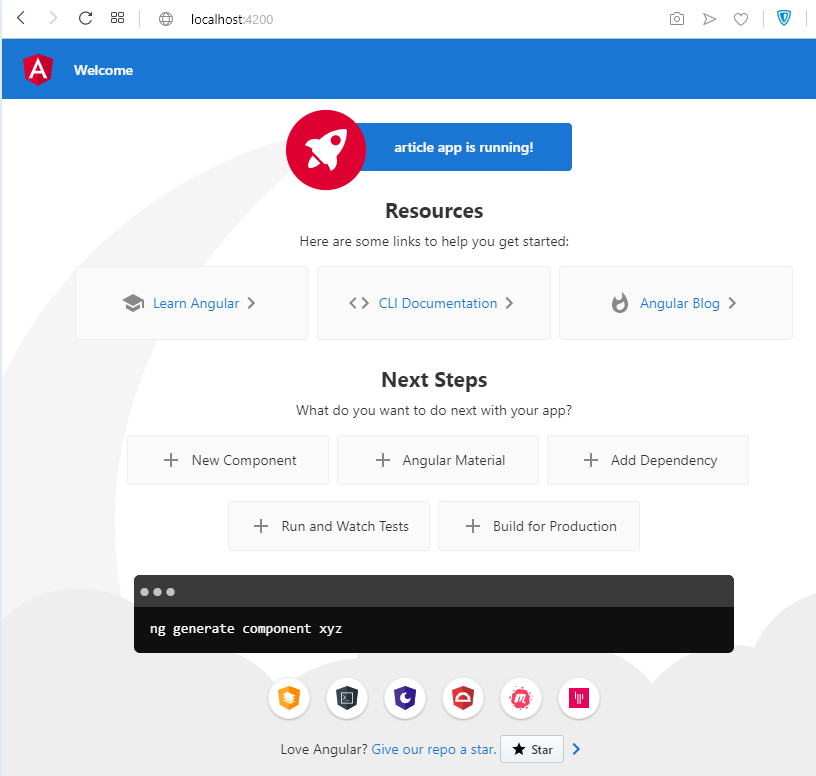
La fenêtre suivante s’affiche et nous en profitons pour invoquer la commande ‘ng help’ afin de savoir quelle commande nous servir pour exécurter nos sources dans le navigateur:



Il s’agit de la commande serve :



Cette commande nous dit que notre application angular disponible par défaut se trouve à l’adresse [http://localhost:4200](http://localhost:4200/)

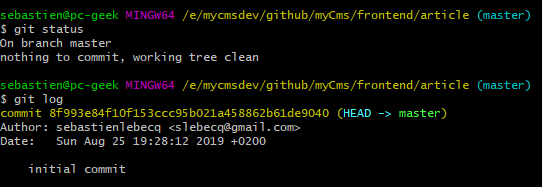


L’information qui nous concerne dans cette page est : article app is running !

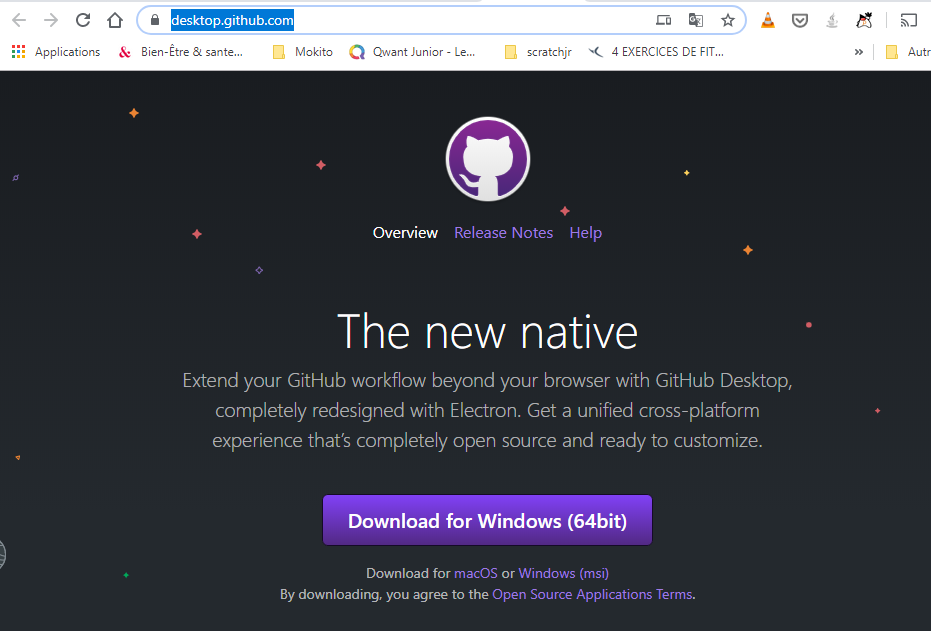
Les liens hypertexte sous le texte Ressouces renvoient vers le site [https://angular.io](https://angular.io/)

Il ne reste plus qu’à synchroniser les sources de notre application Angular ‘article’ avec un répertoire distant hébergé par gitHub sous mon espace sebastienlebecq.

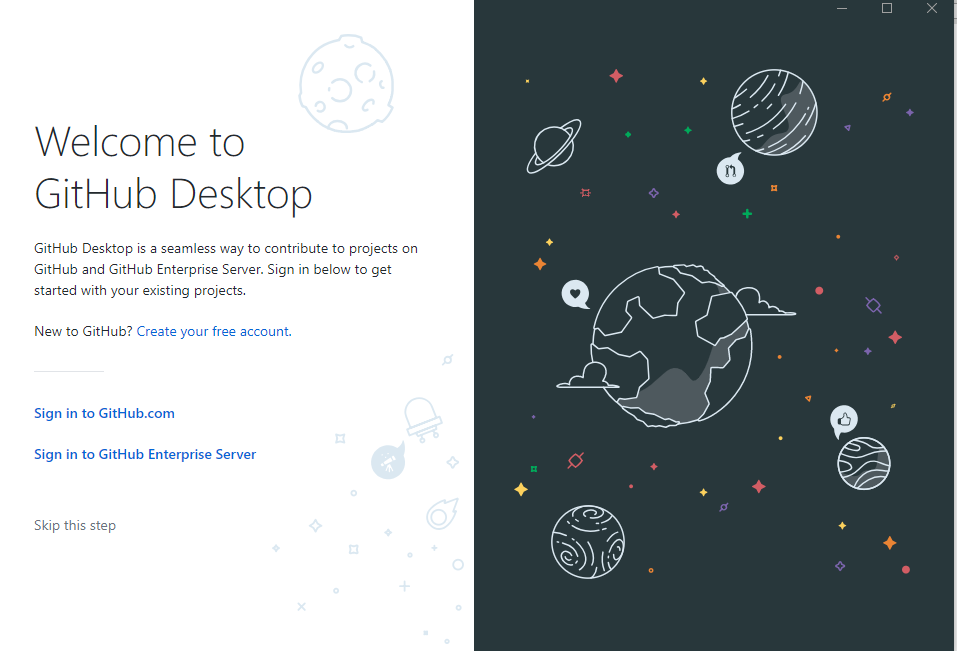
Dans un 1er temps, avec la commande git status, j’interroge Git pour savoir s’il n’y a rien de présent dans le working Directory qui ne serait pas à commiter. Ensuite, je vérifie que le projet a bien été inscrit sous Git par le jeu de a commande git log qui dresse l’historique des derniers commits, en l’occurence le commit initial du projet :

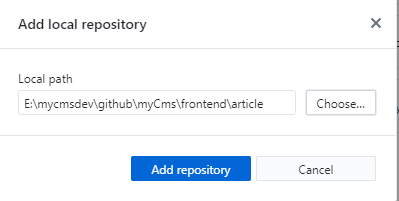
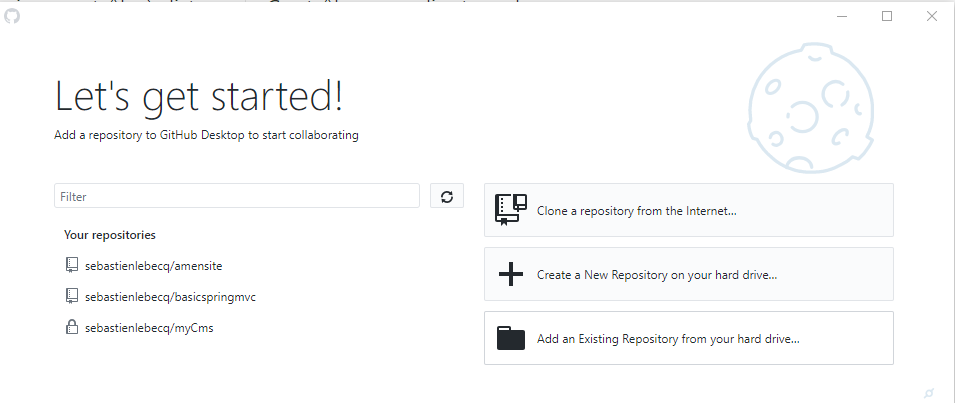


Puisque on est sous windows, téléchargeons github Desktop : <https://desktop.github.com/>

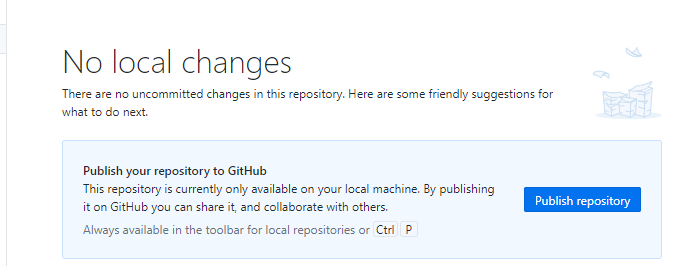


1er écran :

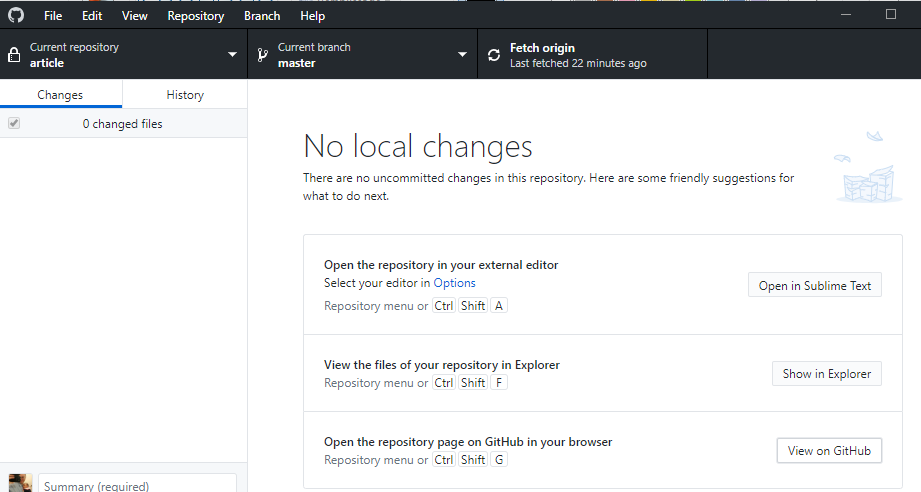




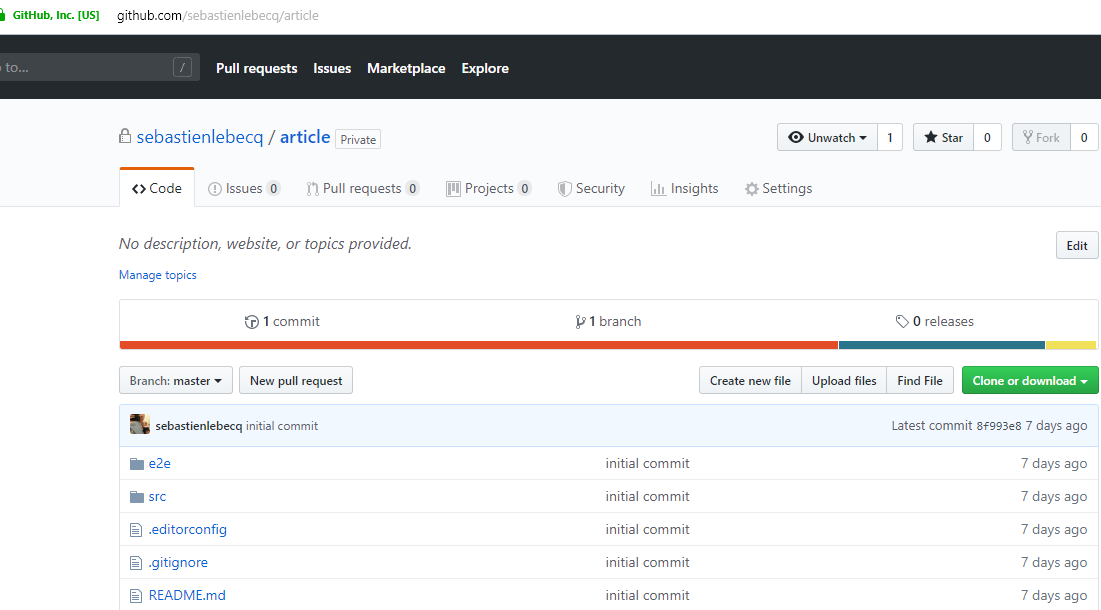
Puis je clique sur le bouton « Add repository » :



Puis je clique sur le bouton « publish repository »



Je clique sur le bouton ‘View On GitHub , je constate la fenêtre suivante dans le navigateur :



Le projet a bien été commit sur le répertoire distant article de github. CQFD