# Tutoriel Coding-Github

Hi !

Ce tuto va t’aider à t’y retrouver dans le code du site dev.microsoft.fr. Il y a des subtilités et des cas particuliers à appréhender mais pas d’inquiétudes au bout de quelques semaines ça devrait aller mieux. Ce tuto est un complément du tuto créé par Hugo nommé « Tutoriel Landing pages », que je conseille de lire juste après celui-ci.

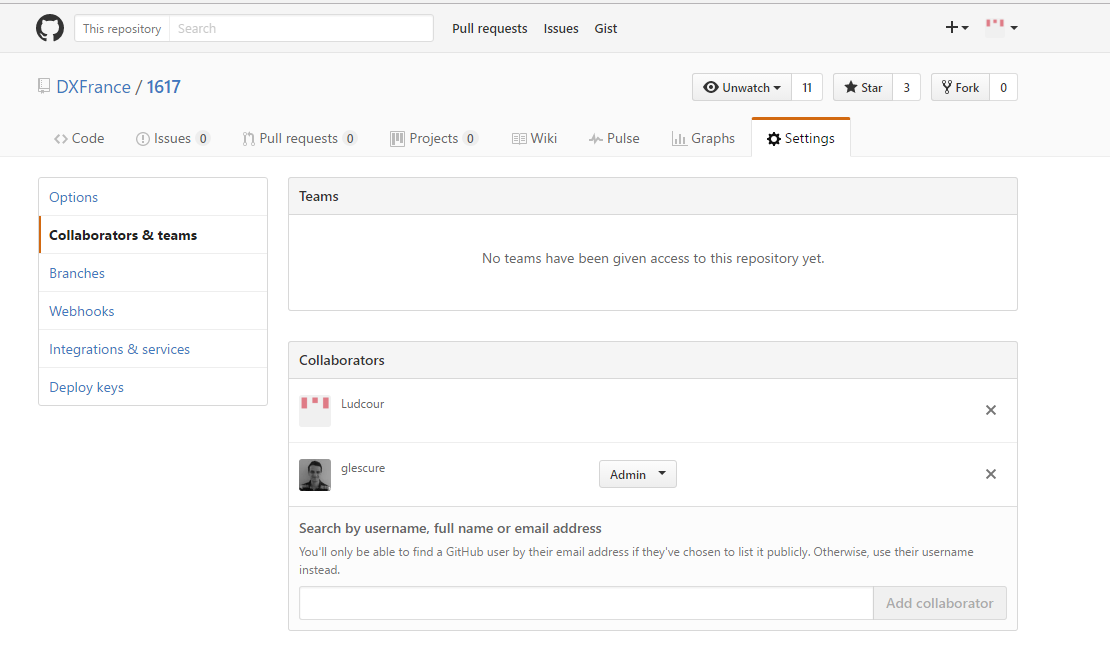
## Getting started with GitHub

L’outil utilisé pour faire des modifications sur le site s’appelle GitHub. Pour info Github est la plateforme de référence en ce qui concerne le développement de projets open-source.

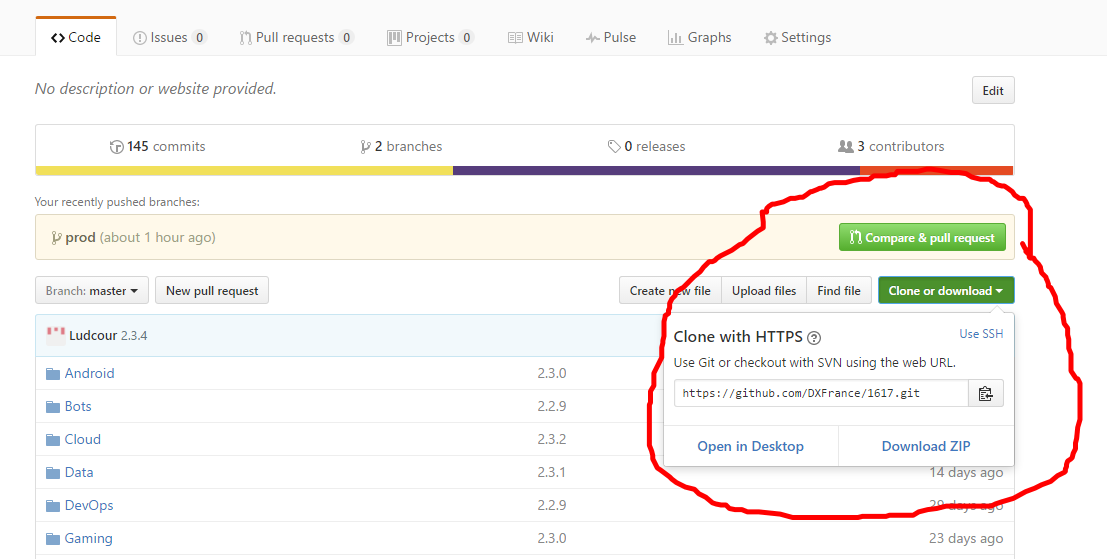
Le site dev.microsoft.fr est donc par définition un projet open-source même si en réalité l’intérêt open source reste limité (pas d’autres contributeurs).

Il y a un certain nombre d’étapes à suivre pour pouvoir commencer à mettre les mains dans le code :

* Se créer un compte sur <https://github.com>
* Demander à Guillaume de vous ajouter en contributeur en tant qu’administrateur



* Une fois invité il vous faut installer l’application desktop de Github (Google is your BF)
* Vous allez maintenant créer une version en local du dossier du site. (cliquer sur open in desktop)



* Suivez les instructions jusqu’à choisir l’emplacement de votre dossier (qui sera par défaut dans un dossier Documents/GitHub.

Bravo ! Vous avez maintenant une version en local du site !

## Utilisation de Github

Comme énoncé précédemment, Github est la plateforme qui nous permet de faire les modifications sur le site. Avant de commencer plus en détail il faut revenir sur la structure et le fonctionnement de base du site :

* Le site dispose de deux branches :
  + La branche de test : <http://starmania-test.azurewebsites.net/> . Cette branche est aussi appelée « Master ». C’est sur celle-ci que vos modifications seront appliquées en premier (plus d’explications plus bas)
  + La branche de prod : <http://dev.microsoft.fr/> . C’est la branche « officielle » qui héberge le site en lui-même.
* Vous avez également en local une version du site que vous avez cloné en suivant les étapes de la section précédente.

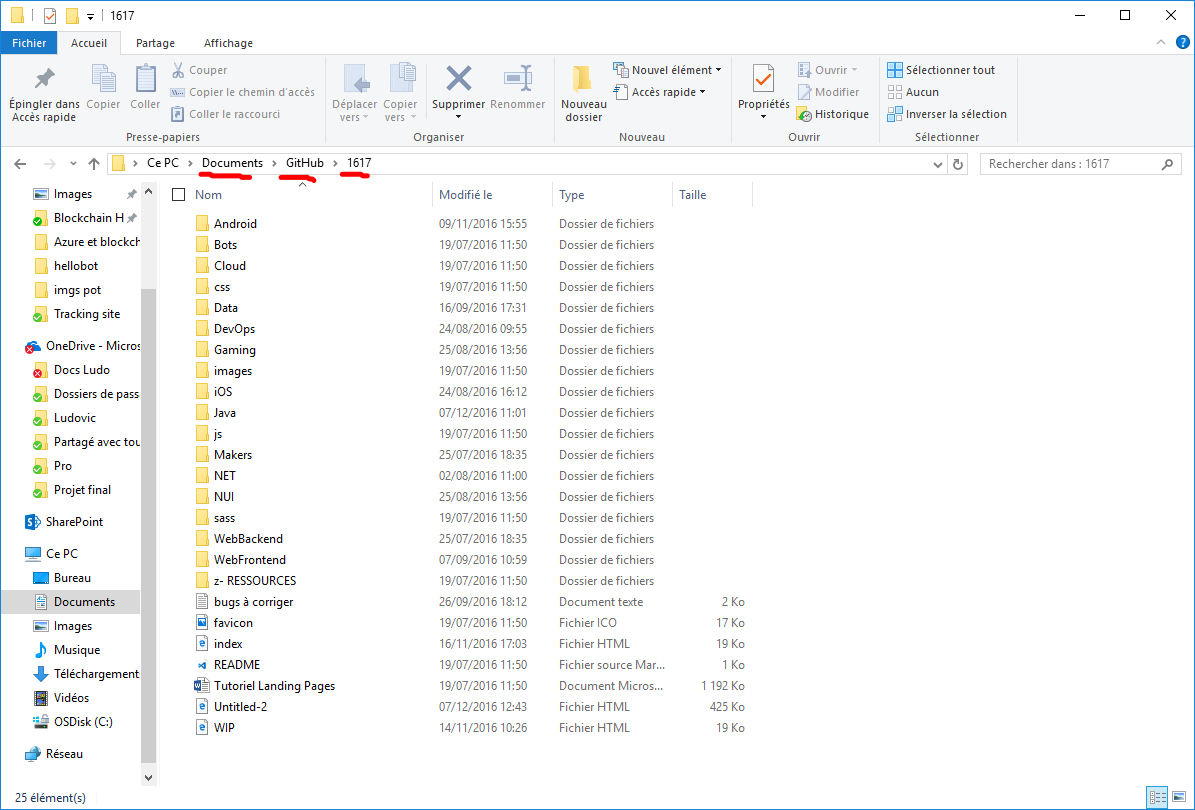
Vous avez donc 3 versions du site :

* La version en local sur laquelle vous travaillerez et apporterez les modifications
* La version de test qui vous permet de montrer à quoi ressemblent les modifications que vous avez effectué et d’avoir un go/nogo de la part des gens avec qui vous travaillez.
* La version prod qui est le site à proprement parler.

On va maintenant voire comment passer d’une branche à l’autre et les spécificités de chaque branche.

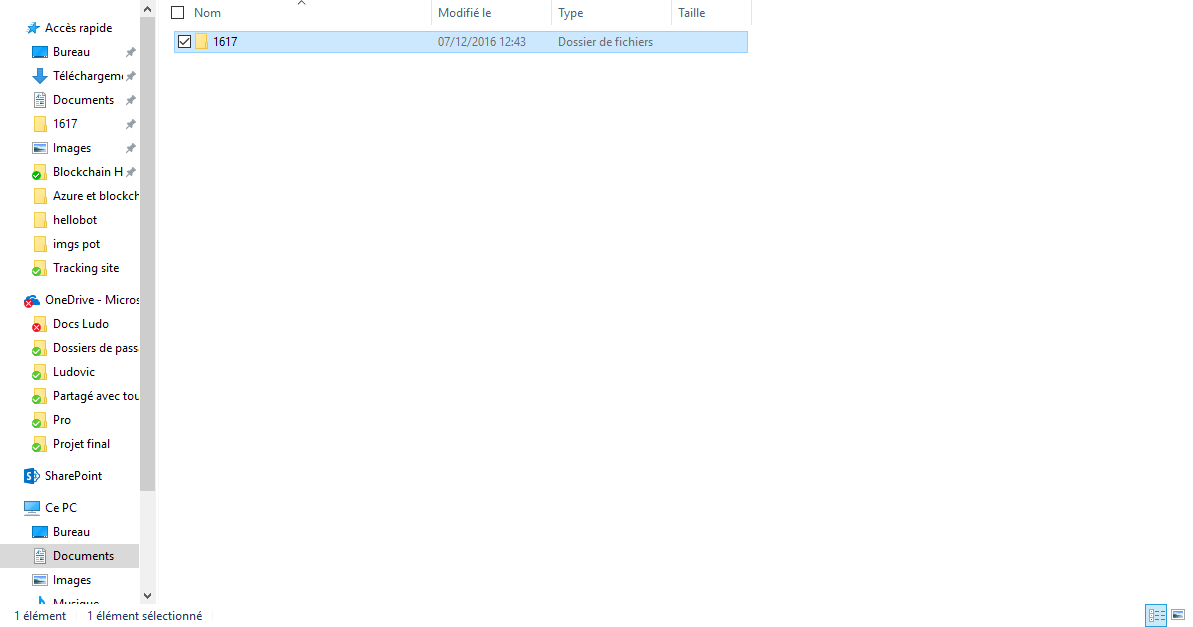
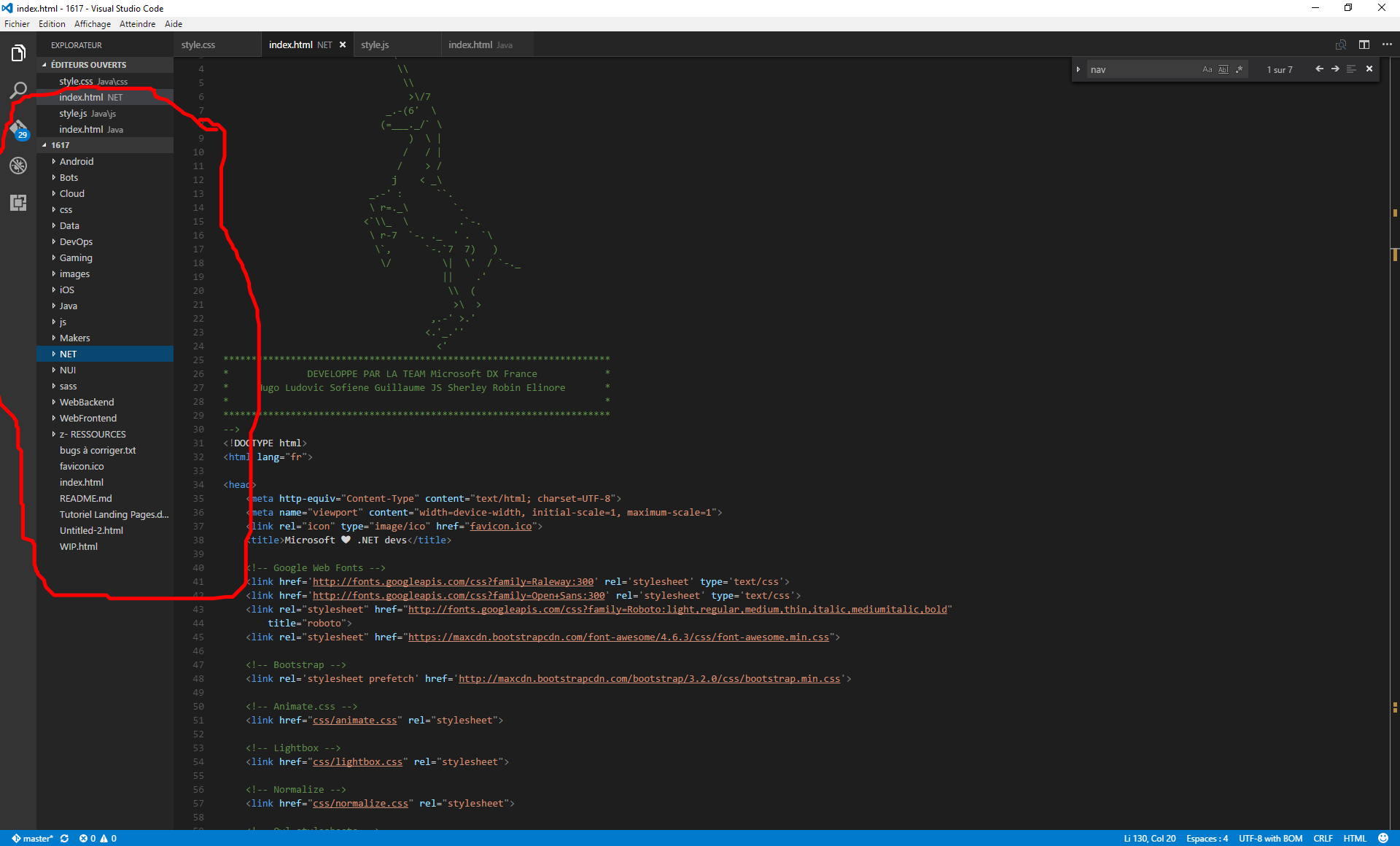
1. La version en local :

Cette version a été créée en suivant les instructions de la section 1. Vous disposez d’un dossier comme suit :



Je ne sais pas quel est votre niveau en codage mais ce que vous voyez juste au dessus c’est « le site », i.e. tous les fichiers/images/ressources de votre site. Je vous conseille d’épingler ce dossier dans vos accès rapides, vous allez en avoir besoin souvent ;). Quand vous souhaiterez ajouter un nouvelle image au site c’est dans ces dossiers qu’il faudra aller en placer une copie.

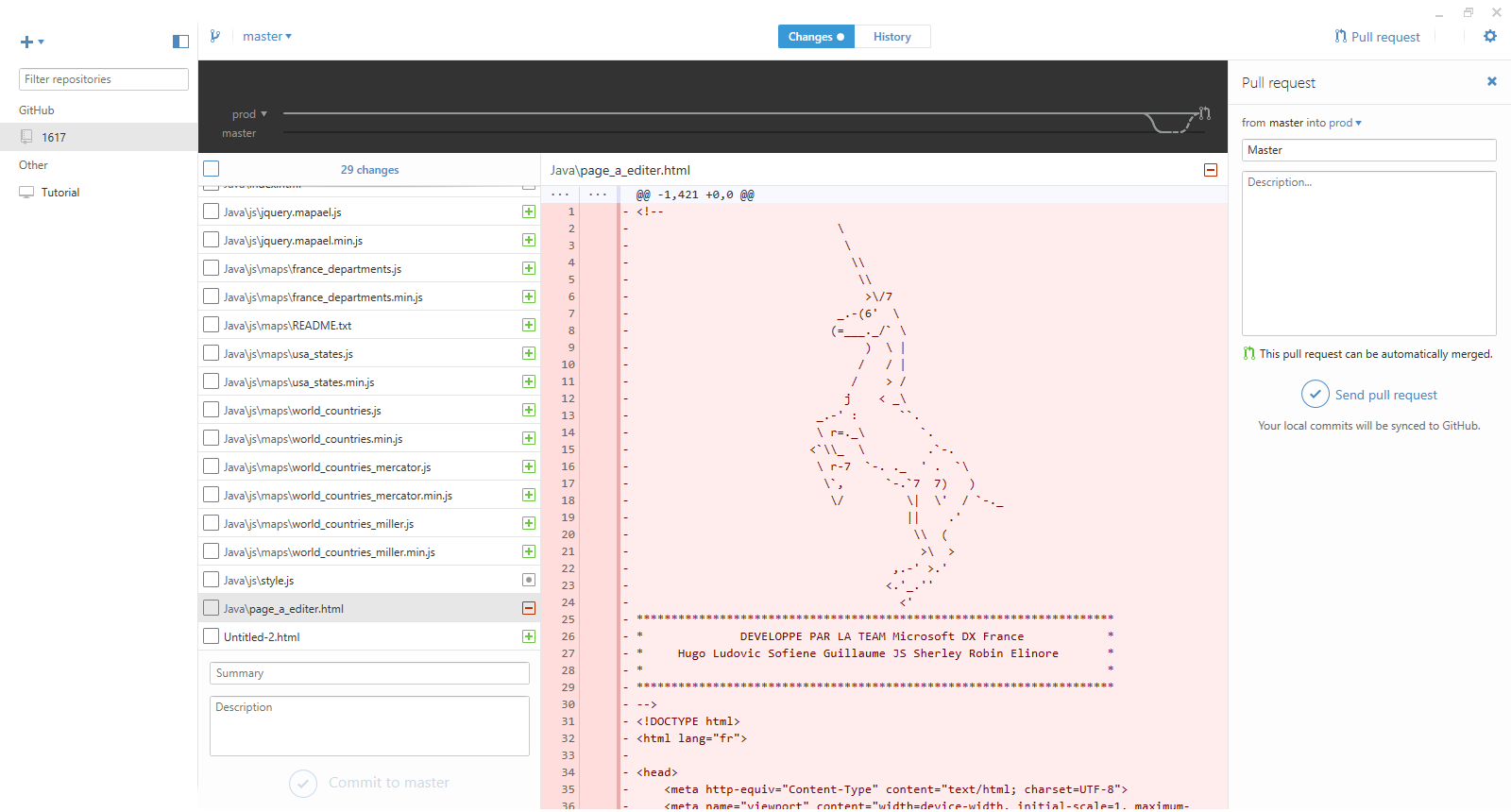
Pour effectuer des modifications sur le code voici la manipulation :

* + Téléchargez visual studio code
  + Faites un clic droit sur le dossier parent « 1617 » (image ci-dessous) et cliquez sur « Open with Code »
  + 
  + Vous avez maintenant tout ce dont vous avez besoin directement accessible. Toutes les modifications de code auront lieu ici (l’ajout ou la suppression de ressources (images/autres) se fait directement dans le dossier vu ci-dessus).
  + 
  + Modifiez le code et sauvegardez vos changements avec un simple CTRL+S. Se reporter au tuto Landing pages pour les modifications du code.

1. Passage du local à la page de test

Vos modifications sont faites ? Vous les avez testé en local sur chrome ? Nice, il est maintenant temps de mettre ces modifications en ligne.

Pour cela allez dans l’application de bureau GitHub que vous avez téléchargé au tout début. Vous pouvez voir que tous les changements que vous avez effectués sont visibles et ont été pris en compte par Github !

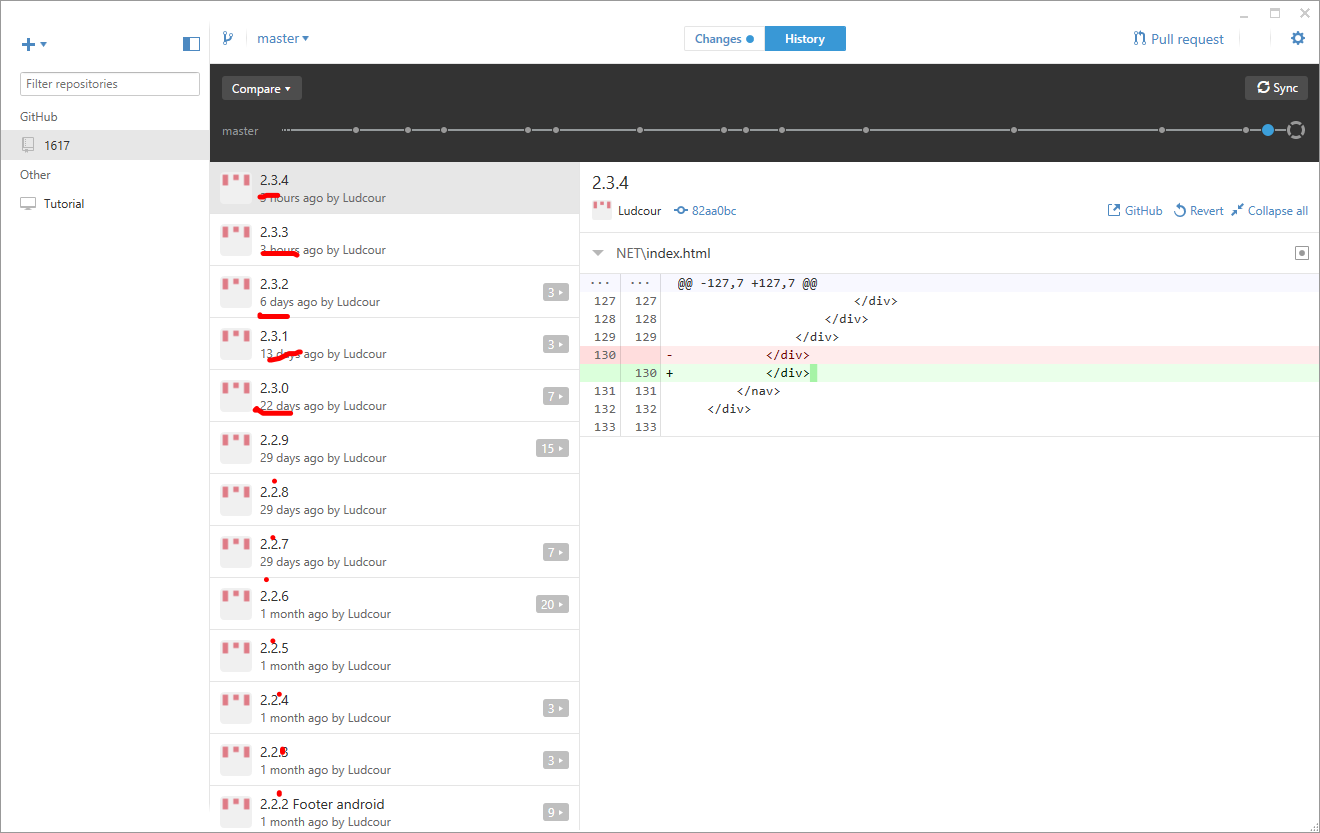


* Sélectionnez les changements que vous voulez mettre en ligne.

PARENTHESE IMPORTANTE POUR VOUS PLUS TARD : Cette possibilité de sélectionner les éléments que vous voulez mettre en ligne est importante. En effet au début vous travaillerez surement sur une seule chose à la fois mais après un certain temps il y aura plusieurs pages différentes avec des modifications pas forcément terminées (comme sur l’image). Faites attention à sélectionner uniquement ce que vous êtes sur de publier définitivement ensuite.

Un exemple pour mieux comprendre : vous travaillez sur la page Java, faites des modifications sans finir tout ce que vous vouliez faire et là on vous demande de faire en priorité une modification sur une autre page. Vous faites les modifications et publiez le tout sur la page de test, sans distinction, et montrez vos modifications prioritaires pour validation. Si c’est validé et que vous voulez publier en prod vous êtes coincé puisque les modifications non terminées de la page Java seront aussi publiées. Ça vous obligera à créer un copie de vos modifications Java et à réintégrer l’ancienne version en test puis de publier le tout, bref une manipulation chiante dont on préfère se passer. Je sais pas si c’est très clair mais en tout cas une chose à retenir : quand vous bossez sur plusieurs pages en même temps faites attention à ce que vous publiez en test !

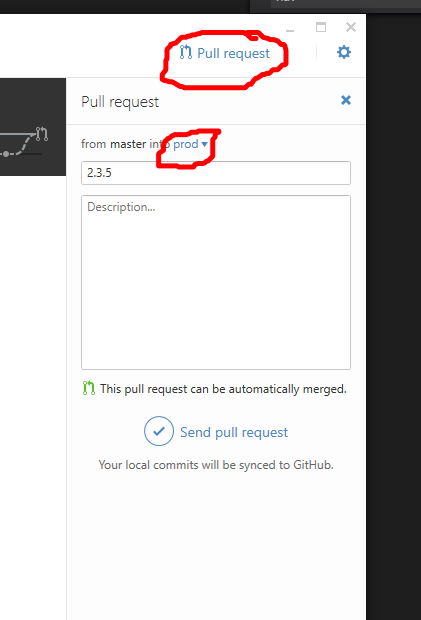
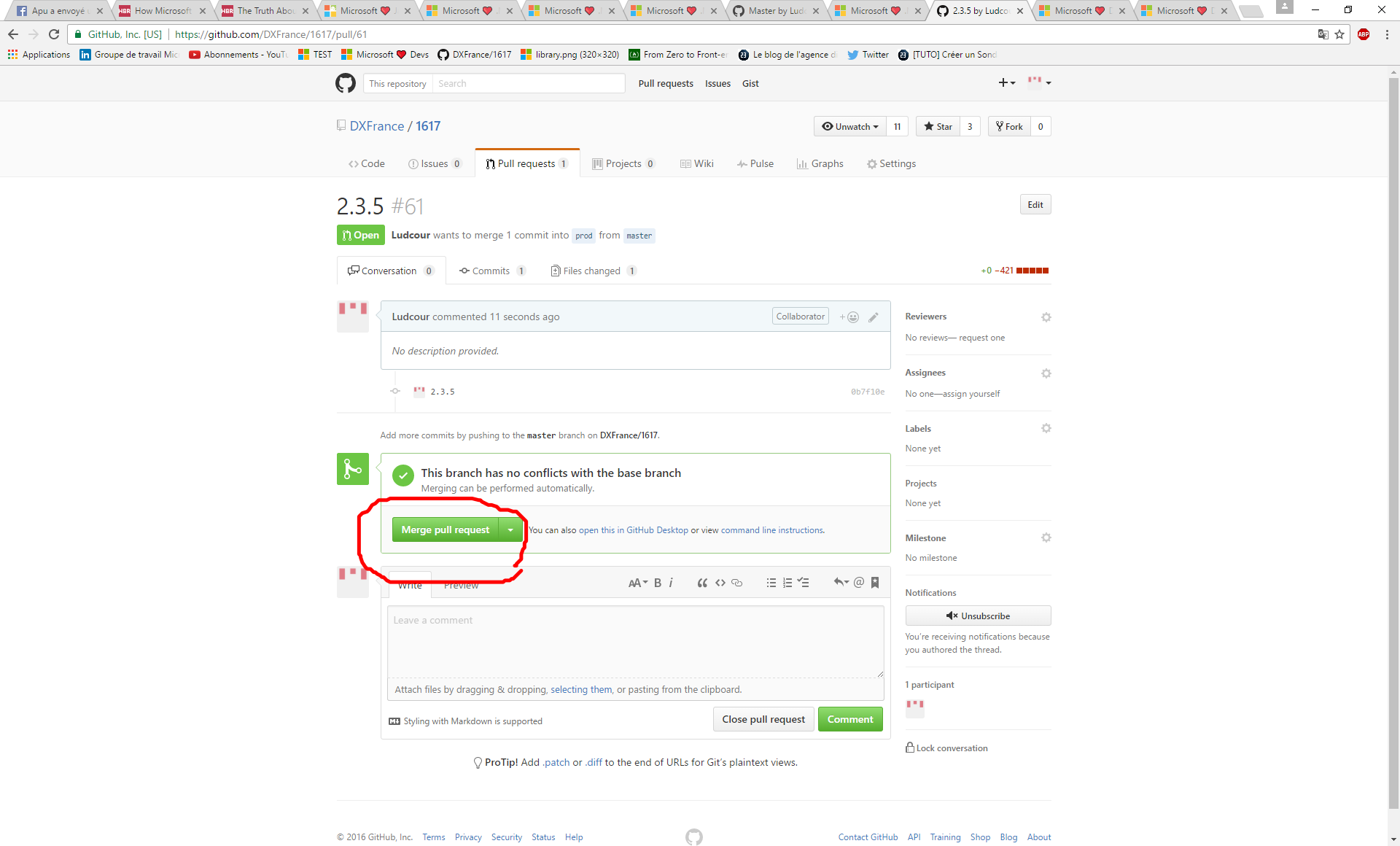
FIN DE PARENTHESE.

* Après avoir sélectionné les modifications que vous souhaitez publier, remplissez la section « Summary » avec le bon numéro de version (que vous trouverez en cliquant sur « History »)
* 
* Cliquez sur « Commit to master ». Le commit est désormais créé, il suffit mainteant d’appuyer sur « Sync » pour que le commit s’upload sur la version de test.
* 
* Vous pouvez aller vérifier maintenant que vos modifications ont bien eu lieu quelques secondes plus tard en vous rendant sur <http://starmania-test.azurewebsites.net/> . Utilisez un CTRL+F5 pour rafraichir la page (et supprimer le cache). CTRL+F5 est une sorte de « hard refresh ».

1. Passage de la version de test à la version définitive

Le passage de la version test à la version définitive, que l’on appelle aussi « passer en prod » depuis la branche « master » (test), fonctionne sur un mécanisme de duplication. L’ensemble de la branche master (test) va être copiée et va venir écraser la version actuelle en prod. On appelle cette opération un pull request.

Le pull request :

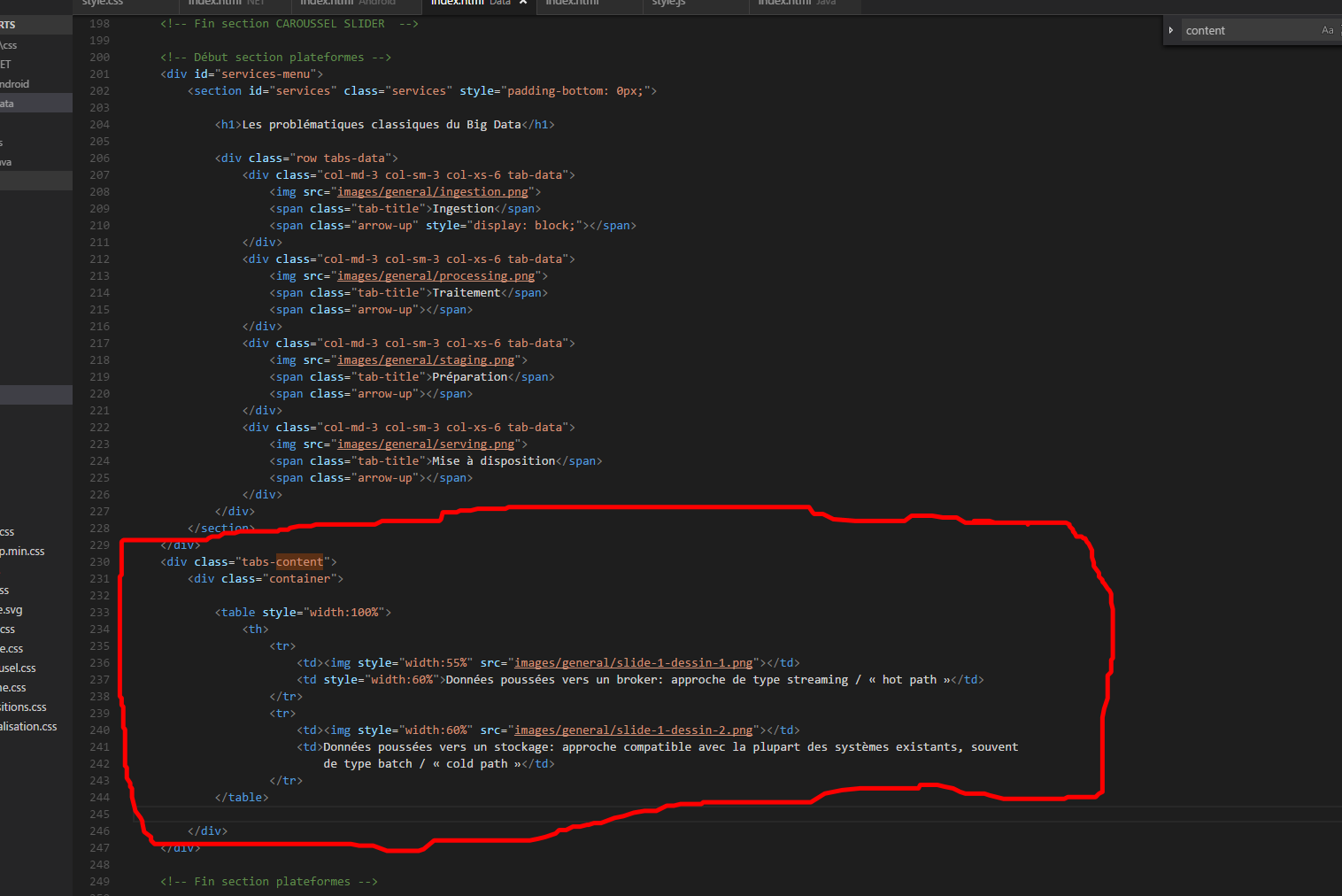
* Dans Github (desktop) cliquez sur « Pull request » et paramétrez « From Master into Prod ». Ce qui signifie de la page test à la page en prod. Puis cliquez sur Send pull request.
* 
* Les pull request sont des propositions d’ajout de la part d’un contributeur. Vu que vous êtes admin vous allez vous-même valider votre pull request en vous rendant sur <https://github.com/DXFrance/1617/pulls> où votre request apparait.
* Après avoir cliqué sur le pull request il suffit de cliquer sur « merge pull request » puis sur « confirm merge ».
* 
* Après une trentaine de secondes environ le merge devient effectif et vous pouvez aller sur <http://dev.microsoft.fr/> pour voir vos changements définitifs (sans oublier de fair CTRL+F5)

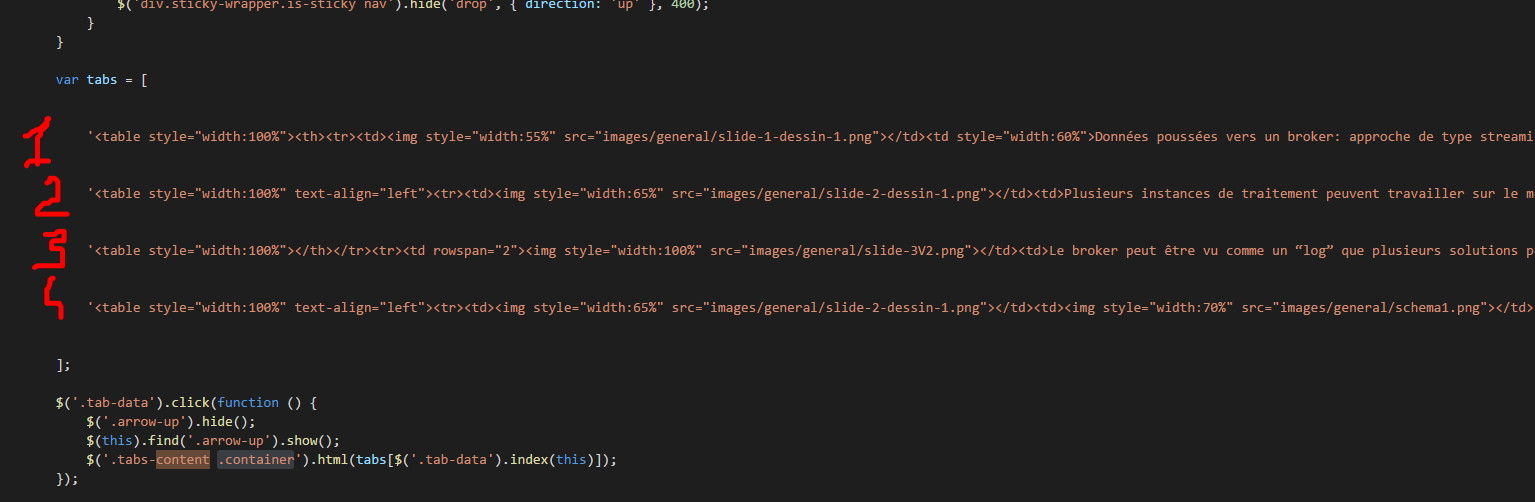
Et voilà ! Vos changements sont en ligne !

Ces trois étapes se déroulent toujours dans cet ordre. Il n’est pas possible de passer directement de la version locale à la version en prod. Néanmoins, on peut répéter plusieurs fois les deux premières opérations sans forcément publier derrière en prod. (exemple : si la modification que vous montrez en test n’est pas validée vous pouvez faire un nouveau « commit » pour le faire valider derrière. Vous ne passerez à la troisième étape que quand vous êtes ok avec cette version.

## Cas particuliers dans le développement des pages

Certaines pages ont en leur sein des particularités concernant leur coding.

* Java : La page Java possède une carte. Les éléments de cette carte sont définis pas un script facilement reconnaissable dans le fichier index.html de la page. La ressource dont vous pouvez avoir besoin est la localisation des villes. Il suffit d’extraire du code de cette page <https://rawgit.com/neveldo/jQuery-Mapael/2.0.0/examples/basic/legend_plotted_cities.html> les villes qui vous intéressent.
* Data : le display de la page est assez différent. La partie cliquable fonctionne avec une base de html qui fait appel a du javascript. 
* La partie encadrée en rouge est ce contenu de la partie « ingestion ». Les autres contenus sont appelés via le fichier style.js (tout en bas)



Je conseille d’éditer dans un fichier séparé ces éléments puis de les remettre copier à nouveau en supprimant tous les espaces (car c’est du HTML inséré dans du javascript)

## Other tips :

Le chargement d’une page en local est souvent long (environ 25 secondes). Cela ne m’a pas toujours fait ça mais je n’ai jamais su trouver pourquoi. En tout cas rien d’inquiétant, cela se charge rapidement quand c’est online.