

Gestion de versions et développement collaboratif



Le portfolio d'Amélia

par José GIL



Sommaire

1. Installation et configuration de NetBeans.....	2
2. Création du projet sous NetBeans	3
3. Création d'un dépôt.....	3
4. Création d'un projet individuel.....	4
5. Gestion de version avancée.....	27
6. Utilisation avancée de GitHub.....	65

1. Installation et configuration de NetBeans

Si vous ne l'avez pas encore, téléchargez la dernière version de NetBeans (version 11 lors de la rédaction de ce support) à l'adresse suivante : <https://netbeans.apache.org/download/>

Plusieurs téléchargements sont proposés, téléchargez les **"binaries"** (exécutables prêts à être utilisés sans installation) :

Downloading Apache NetBeans (incubating) 11.0

Apache NetBeans (incubating) 11.0 was announced on April the 4th, 2019. See [Apache NetBeans 11.0 Features](#) for a full list of features.

Apache NetBeans 11.0 is available for download from your closest Apache mirror. For this release no official installers are provided, please just download the binaries and unzip them.

[Building from source](#)

[Community approval](#)

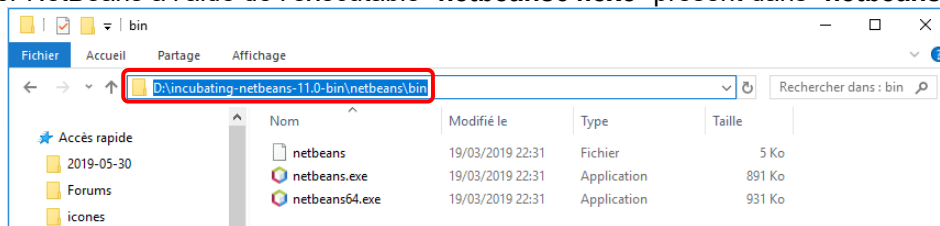
[Release information](#)

[Earlier releases](#)

- Source: [incubating-netbeans-11.0-source.zip](#) (SHA-512, PGP ASC)
- Binaries: [incubating-netbeans-11.0-bin.zip](#) (SHA-512, PGP ASC)
- Javadoc for this release is available at <https://bits.netbeans.org/11.0/javadoc>

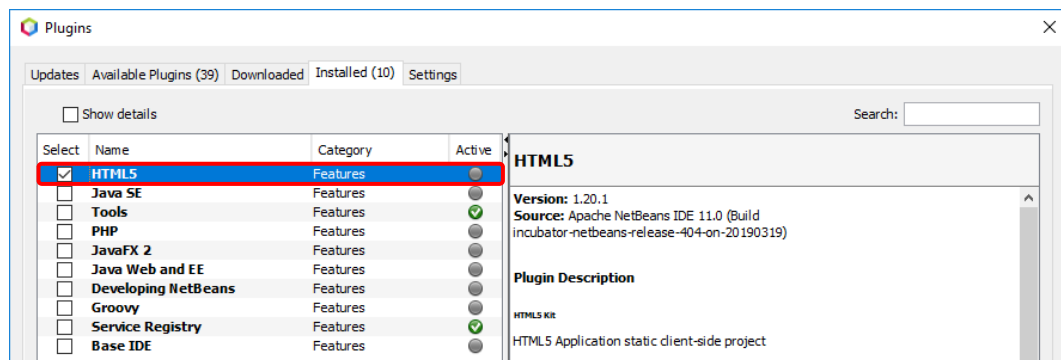
Officially, it is important that you **verify the integrity** of the downloaded files using the PGP signatures (.asc file) or a hash (.sha512 files). The PGP keys used to sign this release are available [here](#).

Extraire ensuite cette archive dans un chemin dépourvu d'espaces ou d'accents (à la racine de C:\ ou de D:\ par exemple) et lancer NetBeans à l'aide de l'exécutable **"netbeans64.exe"** présent dans **"netbeans\bin"** :



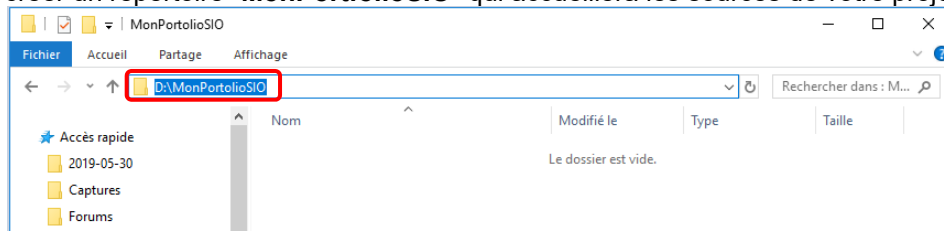
Vous aurez à vous en resservir, n'hésitez donc pas à faire un raccourci (Clic droit ► Envoyer vers ► Bureau).

Une fois lancé, rendez-vous dans le menu **"Tools ► Plugins"**, puis dans l'onglet **"Installed"**, activez le plug-in **"HTML5"** :

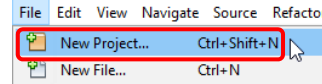


2. Création du projet sous NetBeans

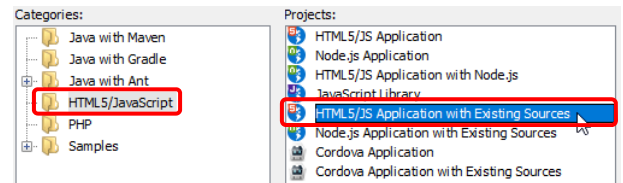
Commencez par créer un répertoire "MonPortfolioSIO" qui accueillera les sources de votre projet :



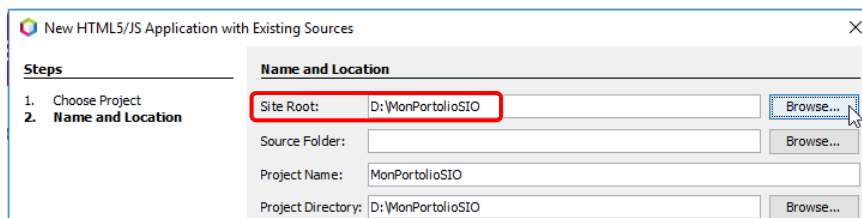
Dans NetBeans, et dans le menu "File", choisir "New Project..." :



Dans la fenêtre de sélection du type de projet, choisissez la catégorie "HTML5/JavaScript", et le projet de type "HTML5/JS Application with Existing Sources", puis cliquez sur "Next" pour passer à l'étape suivante :

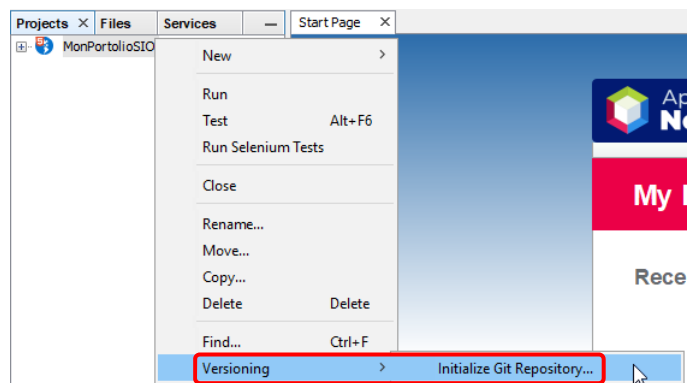


Dans la fenêtre de nommage du projet, cliquez sur "Browse" et sélectionnez le répertoire que vous avez créé. Le "Project Name" est automatiquement nommé avec le nom du répertoire, cliquez sur "Finish" pour terminer la création du projet :

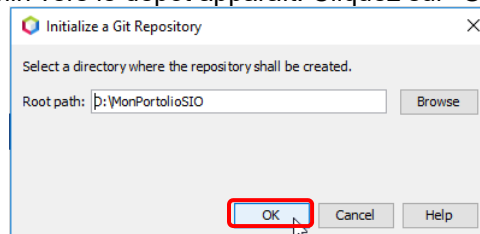


3. Création d'un dépôt

Dès la création du projet, nous allons indiquer à NetBeans que nous souhaitons versionner notre développement. Pour cela, dans l'onglet "Projects" vous allez cliquer sur le nom du projet avec le bouton droit, puis choisir "Versioning" et enfin "Initialize Git Repository" :

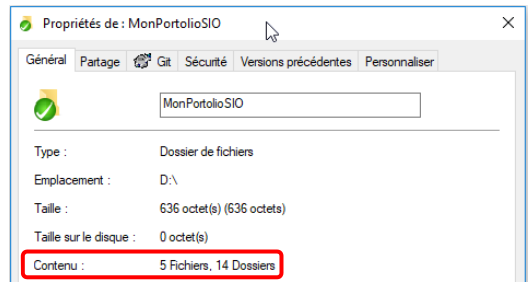


Une fenêtre de confirmation du chemin vers le dépôt apparaît. Cliquez sur "OK" :



On peut voir à présent visuellement que le projet est versionné : un "badge" bleu en forme de base de données apparaît à côté du projet.

Bien que l'opération précédente ne semble n'avoir rien fait, il se trouve qu'en réalité un nouveau dossier caché nommé **".git"** a été ajouté. C'est ce dossier qui contiendra toutes les informations liées à la gestion de versions. Si votre explorateur est configuré pour afficher les fichiers et dossiers cachés, vous l'avez vu, sinon il vous suffit de cliquer avec le bouton droit et de choisir **"Propriétés"** dans le menu contextuel pour voir qu'en réalité le dossier contient maintenant 5 fichiers et 14 dossiers.

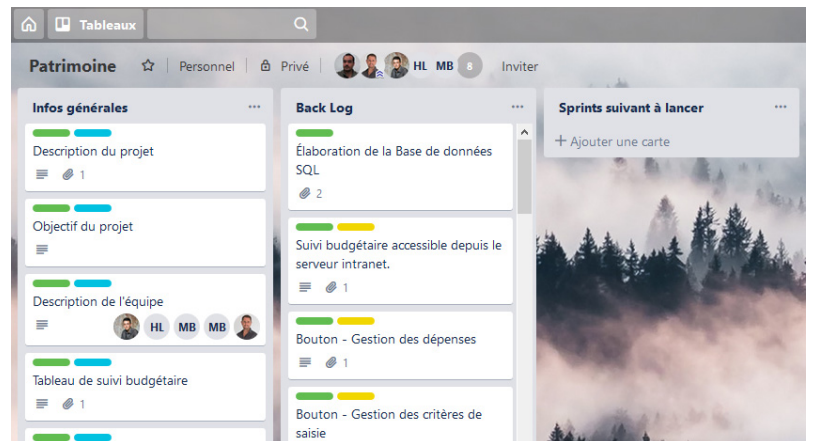


4. Création d'un projet individuel

Nous allons faire comme si l'on démarrait un projet d'application web. Même si l'objectif ici est de vous faire travailler sur la gestion de version avec Git, pas de faire du "vrai développement" (vous disposez des autres cours pour cela) nous allons tout de même joindre l'utile à l'agréable en commençant un portfolio qui pourra vous resservir ensuite pour les épreuves E4 et E6 et plus généralement pour vous permettre de disposer d'une identité numérique (recherche de stages, d'emplois...). Mais avant toute chose, il est important de réfléchir à la démarche.

4.1 La planification au quotidien

Il existe de nombreux logiciels pour planifier les projets simplement. Par exemple la plateforme [Trello.com](https://trello.com), très utilisée en entreprise, propose de s'organiser en suivant le principe de la méthode Kanban grâce à des listes (par exemple "À faire", "En cours" et "Fait", comme cela a été abordé dans le cours), chacune comportant ensuite des cartes ("kanban" en japonais) avec le détail des tâches :



TRAVAIL À FAIRE

Créez un compte sur la plateforme [Trello.com](https://trello.com) et appropriez-vous l'application en commençant par exemple à créer un premier "tableau" pour vous aider à organiser votre quotidien. On peut imaginer une première liste avec les devoirs à faire où vous placerez une carte par devoir en paramétrant une date limite. Pourquoi par une autre liste "Dette technique" avec les nouveaux concepts à approfondir qui vous permette de vous mettre à jour sur une nouvelle technologie ou un nouvel outil avec des cartes comportant le nom du concept et un lien vers une vidéo à voir dès que vous avez quelques minutes à y consacrer. Etc.

Les bienfaits d'un tel outil sont nombreux. Tout d'abord il vous permet de lutter contre la procrastination avec par exemple une liste "En cours", ou "À faire". Le tout étant de placer dans cette liste les tâches que l'on s'engage à réaliser dans la journée (c'est pourquoi il faut faire attention à ne mettre que des objectifs atteignables) et d'autre part il sert aussi d'outils d'autovalorisation et d'amélioration de l'estime de soi.

En effet, sans mesure de l'activité, il est assez courant qu'en fin de journée on ait l'impression de n'avoir rien fait. En ajoutant une liste "Fait", et en y transférant pendant la journée les tâches que l'on termine, cela permet de faire un bilan positif en fin de journée. Un moment qu'il est important de ritualiser : Voir tout ce que l'on a réussi à faire, et ensuite supprimer ces tâches pour laisser un tableau prêt à être réutiliser le lendemain.

4.2 La planification dans GitHub

Après avoir mis en œuvre Git, ce TP a également pour objectif de vous faire mettre en œuvre la plateforme GitHub. Comme on l'a vu dans le cours cette forge logicielle offre de nombreux outils. Deux outils nous intéressent plus précisément ici :

- La "**Gestion de projet GitHub**" : Ce service nous permet de disposer d'un Kanban que l'on va pouvoir lier directement à notre projet pour automatiser certaines opérations de suivi (par exemple pour suivre l'évolution d'une tâche de sa création, à sa prise en charge et jusqu'à sa clôture).
- Les "**Pages GitHub**" : Ce service vous permet d'héberger directement sur GitHub un site web fait à l'aide de pages statiques en HTML/CSS/JS. Nous allons donc nous en servir pour héberger notre portfolio.

■ Création d'un compte GitHub

Rendez-vous sur <https://github.com> et si vous n'en avez pas encore, créez un compte :

- Saisissez un nom d'utilisateur
- Saisissez un mail valide
- Saisissez un mot de passe robuste

Et cliquez ensuite sur "**Sign up for GitHub**" pour valider la création du compte.

Username: AmeliaPilou ✓

Email: amelia.pilou@gmail.com ✓

Password: [masked] ✓

Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. [Learn more.](#)

Sign up for GitHub

Validez ensuite la souscription d'un compte gratuit en cliquant sur "**Continue**" :

Welcome to GitHub

You're a few steps away from building better software. @AmeliaPilou.

Completed: Set up your account

Step 2: Choose your subscription

Step 3: Personalize your experience

Choose your subscription

With tools developers love and the world's largest open source community, there's no wrong choice.

Free

The basics of GitHub for every developer

\$0 per month

Includes:

- ✓ Unlimited public and private repositories
- ✓ 3 collaborators for private repositories
- ✓ Issues and bug tracking
- ✓ Project management

Pro

Pro tools for developers with advanced requirements

\$7 per month (view in EUR)

Includes:

- ✓ Unlimited public and private repositories
- ✓ Unlimited collaborators
- ✓ Issues and bug tracking

Dans les choix proposés dans l'écran suivants, choisissez le niveau de programmation que vous voulez, Cochez ensuite les cases "**Hosts a project (repository)**" et "**Creating a website with GitHub Pages**" et cliquez ensuite sur "**Submit**" :

Welcome to GitHub

You'll find endless opportunities to learn, code, and create. @AmeliaPilou.

Completed: Set up a personal account

Step 2: Choose your subscription

Step 3: Tailor your experience

What is your level of programming experience?

☐ None—I don't program at all

☒ New to programming

☐ Somewhat experienced

☐ Very experienced

What do you plan to use GitHub for? (Select up to 3)

☐ Learning to code

☐ Learning Git and GitHub

☒ Host a project (repository)

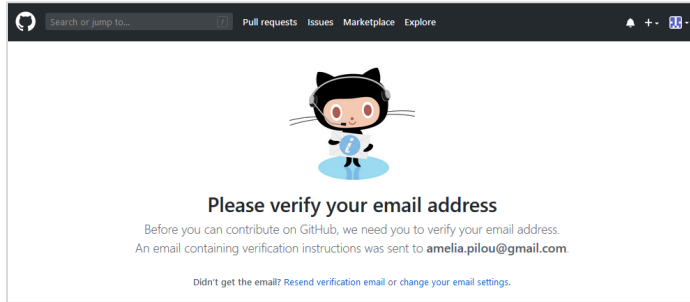
☒ Creating a website with GitHub Pages

☐ Collaborating with my team

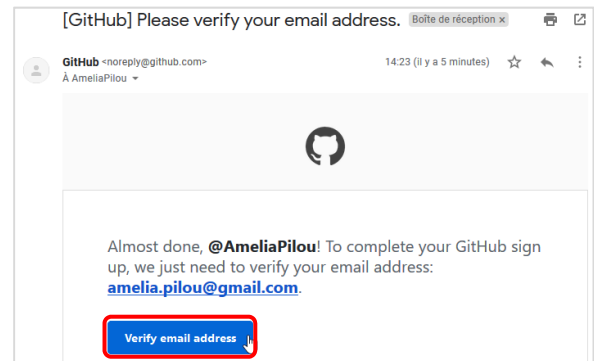
☐ Finding a project to contribute to

☐ School work / School-related project

L'écran suivant vous informe qu'un mail vous a été envoyé pour vérifier votre adresse.

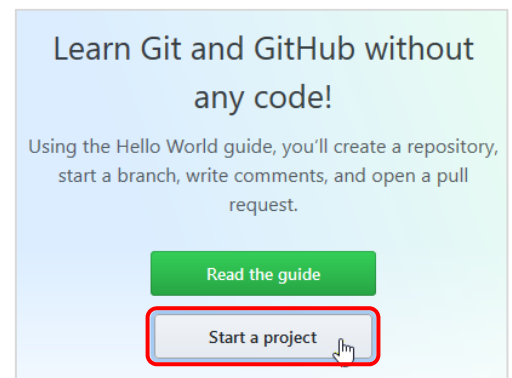


Rendez-vous dans votre boîte de messagerie pour cliquer sur le bouton "Verify email address" dans le message reçu :



Création d'un dépôt GitHub

De retour sur GitHub, un message vous indique que votre email a bien été vérifié. Sur cette même page, on vous invite à démarrer un projet. Cliquez sur **"Start a project"** :



Comme nom de dépôt, saisissez "<votre nom d'utilisateur>.github.io". C'est une condition pour que l'on puisse ensuite utiliser cette adresse pour accéder à l'hébergement de notre site que nous verrons plus loin. L'autre condition est également de laisser le dépôt en accès **"Public"**.

Saisissez également une **"Description"**, et cochez la case **"Initialize this repository with a README"**.

Validez ensuite la création de ce dépôt en cliquant sur le bouton **"Create repository"**.

Dans la page suivante, on peut voir que le dépôt a été créé, et que le contenu saisi dans la description a été transféré dans le fichier **"README.md"**.

TRAVAIL À FAIRE

À l'aide de la plateforme [Dillinger.io](https://dillinger.io) et du site markdownguide.org, initiez-vous à la syntaxe du langage de description Markdown pour améliorer le fichier **"README.md"** afin de mieux présenter le projet de portfolio.

Exemple avancé de mise en page :

40 lines (30 sloc) | 3.64 KB

Raw Blame History

Portfolio d'Amélia PILOU

Objectifs :

- Mettre en oeuvre les technologies web pour créer et héberger mon portfolio



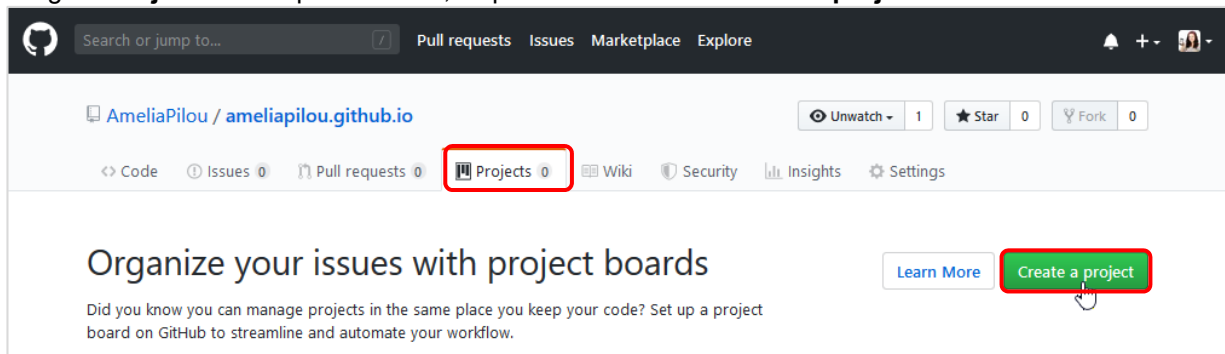
- **NetBeans** : Environnement de développement intégré (IDE) utilisé pour ce projet
- **HTML** : Langage de description utilisé pour concevoir les pages web
- **CSS** : Langage de style utilisé pour mettre en forme les pages web
- **Bootstrap** : Framework graphique utilisé pour faciliter la création de pages web (utilisation d'une base de règles CSS existantes)
- **Git** : Logiciel de gestion de version
- **GitHub** : Forge logicielle en ligne utilisant Git

- Pour l'épreuve E6
 - Je me présente et présente mon parcours, j'affiche mon CV
 - Je parle des projets que j'ai réalisés, je bascule sur les pages correspondantes
 - Je parle des stages que j'ai effectué et des projets auxquels j'ai participé, je bascule sur les pages correspondantes
 - Je présente mes bonnes pratiques :
 - Respect des normes et des standards
 - Auto-formation
 - Mise en oeuvre des tests (unitaires, d'intégration, fonctionnels, ...)
 - Documentation
 - Gestion de version, développement collaboratif et gestion d'incidents
 - Je présente la façon dont j'organise ma veille et la façon dont je m'en sert
- Pour l'épreuve E4
 - J'héberge les documents présentant le contexte de mes PPE
 - J'héberge le code source de mes PPE
 - J'héberge mes productions (compte-rendus, documentations techniques, documentations utilisateurs, ...)
 - J'ajoute des liens vers les applications téléchargeables ou testables en ligne
- Pour mon identité numérique
 - Cela fait un point d'entrée professionnel à donner lorsque je postule à un stage, à une poursuite d'études ou à un emploi
 - Cela permettra de me faire référencer par les moteurs de recherches (là aussi si on me recherche suite à ma rencontre ou à la réception de mon CV)

Création d'un projet GitHub

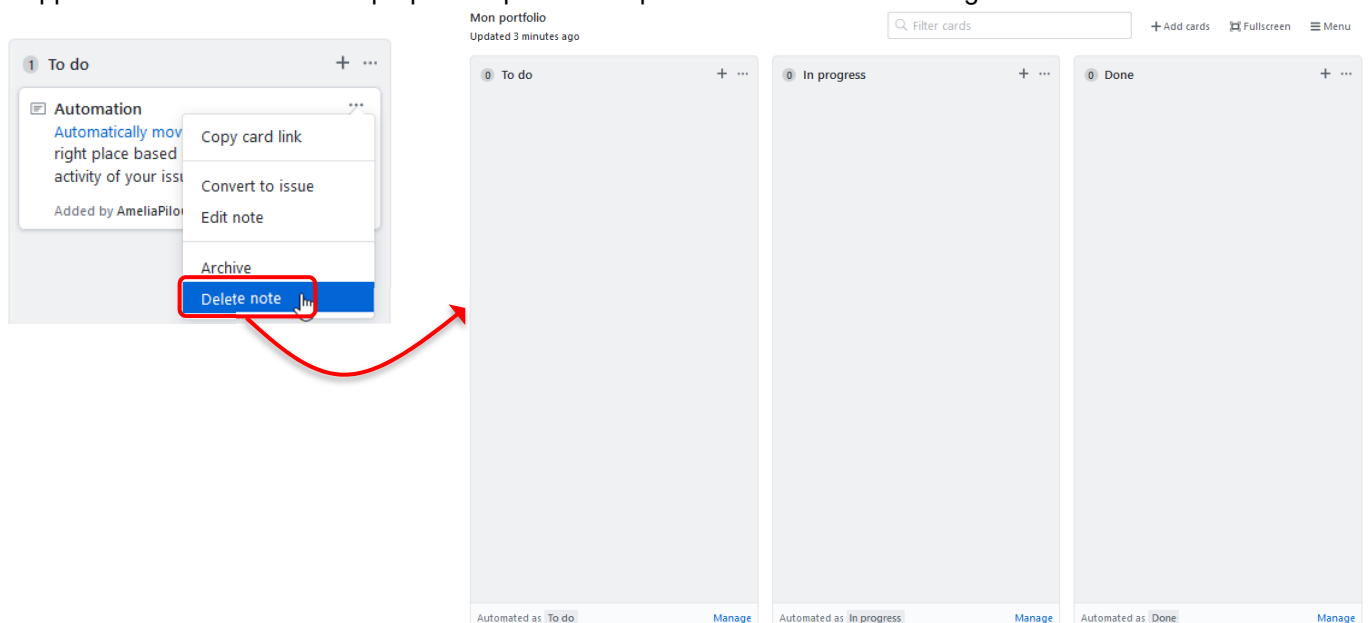
Un **"Projet GitHub"** va nous fournir un Kanban "automatisé" pour que par défaut, chaque ticket d'incident ("**Issue**") soit automatiquement ajouté à la liste "À faire" ("**To do**") et pour qu'ensuite, lorsqu'une tâche est changée de liste, son statut soit automatiquement mis à jour dans le système de tickets.

Dans l'onglet "**Projects**" du dépôt en cours, cliquez sur le bouton "**Create a project**" :

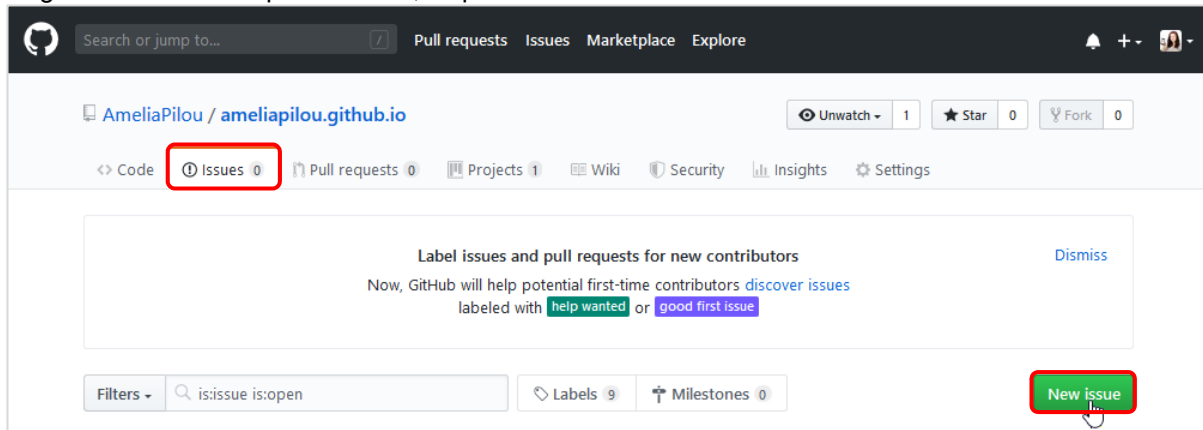


Saisissez un nom de tableau de bord, puis choisissez le modèle ("**template**") "**Automated Kanban**" et cliquez sur le bouton "**Create project**" pour la valider la création :

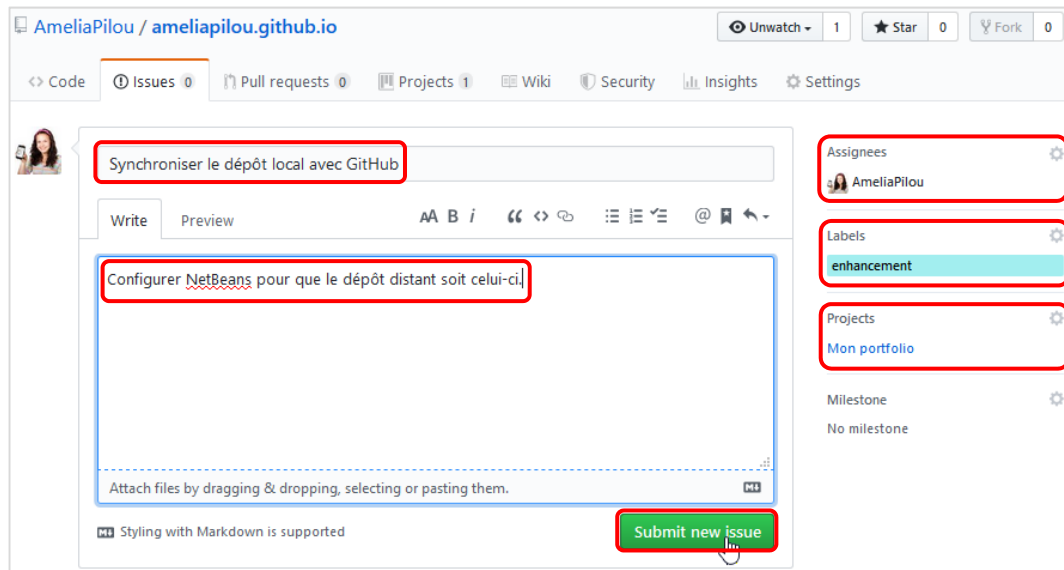
Supprimez les cartes modèles proposées par GitHub pour obtenir un tableau vierge :



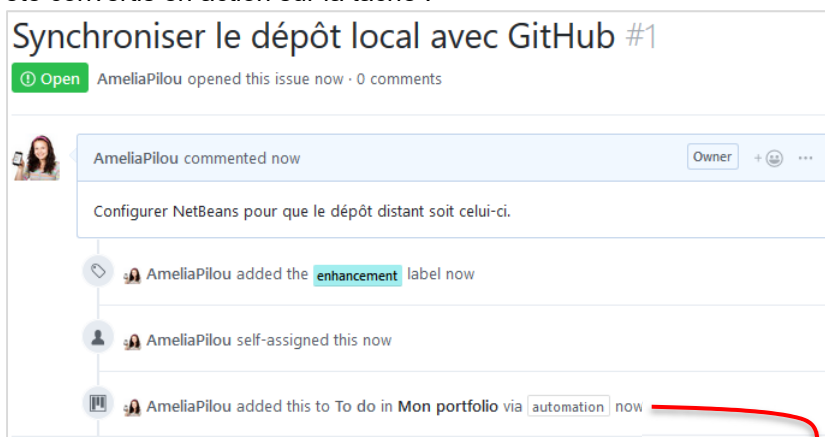
Dans l'onglet "**Issues**" du dépôt en cours, cliquez sur le bouton "**New issue**" :



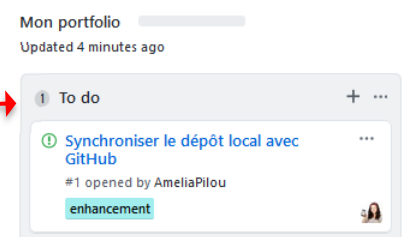
Saisissez le titre "**Synchroniser le dépôt local avec GitHub**" et la description "**Configurer NetBeans pour que le dépôt distant soit celui-ci.**". Assignez-vous cette tâche, choisissez le label "**enhancement**" indiquant une nouvelle fonctionnalité, choisissez le projet en cours "**Mon portfolio**" et cliquez sur le bouton "**Submit new issue**" pour valider ce nouveau "ticket" :



La nouvelle tâche est alors affichée, et l'on peut voir que les différents paramètres saisis pendant la création ont été convertis en action sur la tâche :



En retournant sur votre Kanban, vous devriez voir la nouvelle tâche comme nouvelle carte de la liste "**To do**" :



TRAVAIL À FAIRE

De la même façon que pour cette première tâche, créez les tâches suivantes :

- Créer la page d'accueil
- Ajouter un favicon (mettre un label **"bug"**, ça permet d'en utiliser un différent, et après tout ici on va corriger un dysfonctionnement)
- Ajouter une barre de navigation
- Ajouter un pied de page

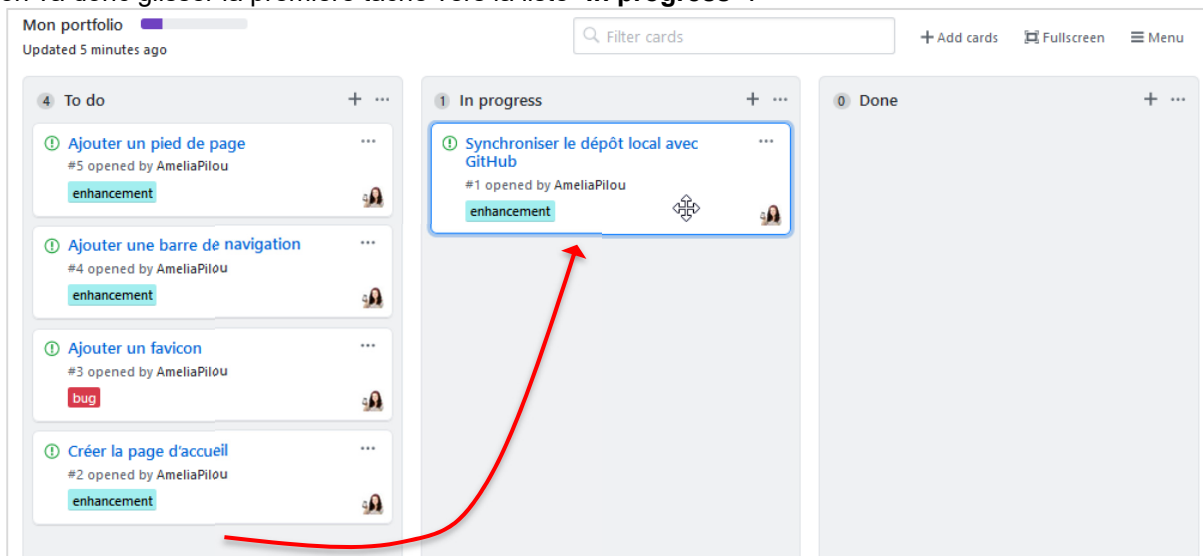
Voilà ce que vous devriez obtenir :

☐	① 5 Open ✓ 0 Closed	Author	Labels	Projects	Milestones	Assignee	Sort
☐	① Ajouter un pied de page enhancement #5 opened now by AmeliaPilou						
☐	① Ajouter une barre de navigation enhancement #4 opened il y a 2 minutes by AmeliaPilou						
☐	① Ajouter un favicon bug #3 opened il y a 4 minutes by AmeliaPilou						
☐	① Créer la page d'accueil enhancement #2 opened il y a 5 minutes by AmeliaPilou						
☐	① Synchroniser le dépôt local avec GitHub enhancement #1 opened il y a 20 minutes by AmeliaPilou						

4.3 La prise en charge en local

Mettre à jour le Kanban

On va prendre en charge la tâche qui consiste à lier notre dépôt local au dépôt GitHub. Dans notre Kanban en ligne, on va donc glisser la première tâche vers la liste **"In progress"** :



Configurer notre dépôt local

Rendez-vous dans le menu **"Git ► Repository ► Open Global Configuration"** et personnalisez les informations : name et email :



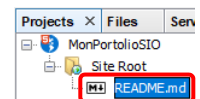
Récupérer le contenu de notre dépôt distant ("remote") GitHub

Rendez-vous ensuite dans le menu **"Git ► Remote ► Pull"**, choisissez l'option **"Specify Git Repository Location"**, et saisissez l'adresse de votre dépôt dans le **"Repository URL"**, votre **"email"** et votre **"mot de passe"** puis cliquez sur **"Next"** :

The first screenshot shows the **Git** menu with **Remote** > **Pull...** selected. The second screenshot shows the **Pull from Remote Repository** dialog. In the **Steps** section, **1. Remote Repository** is selected. In the **Remote Repository** section, **Specify Git Repository Location** is chosen. The **Repository URL** is set to `https://github.com/AmeliaPilou/ameliaPilou.github.io.git`. The **User** is `ju@gmail.com` and the **Password** is masked. The **Remote Branches** section shows **master -> origin/master**. The third screenshot shows the **Clone or download** dialog with the **Clone with HTTPS** option selected and the **Repository URL** pasted.

L'adresse du dépôt peut être récupérée directement sur la page d'accueil du dépôt GitHub :

Après l'opération, vous devriez avoir le fichier **README.md** présent dans votre projet en local :



Rendez-vous dans le menu **"Git ► Repository ► Repository Browser"**, vous pouvez alors vérifier que le dépôt GitHub est bien déclaré :

The first screenshot shows the **Git** menu with **Repository** > **Repository Browser** selected. The second screenshot shows the **Git Repository Browser** dialog. The **Remotes** section shows the **origin** remote with the URL `https://amelia.pilou%40gmail.com@github.com/AmeliaPilou/ameliaPilou.github.io.git`.

TRAVAIL À FAIRE

Mettez à jour le Kanban pour déclarer la tâche comme terminée en la glissant de la liste **"In progress"** à **"Done"**. Cliquez ensuite sur la tâche pour afficher ses détails, et cliquez sur le bouton **"Close issue"** :

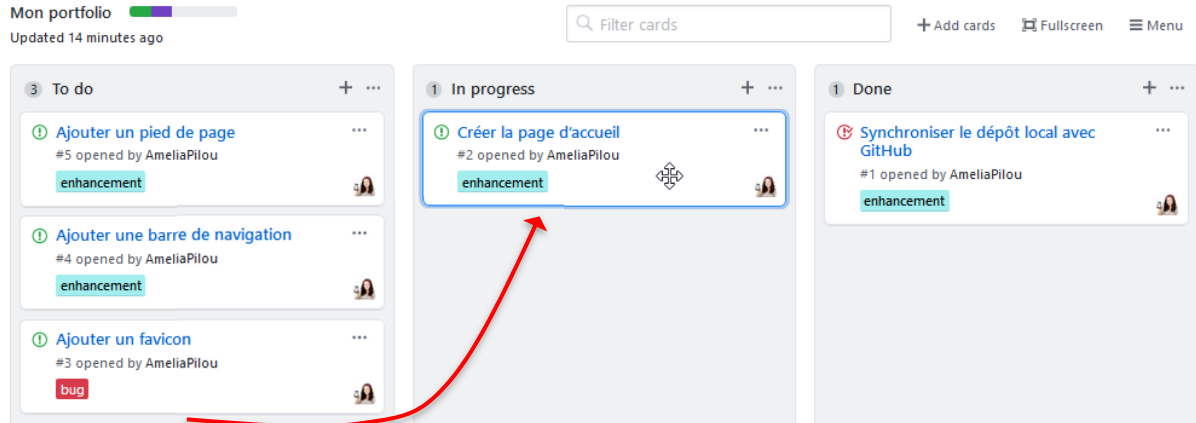
The first screenshot shows the Kanban board with the **Done** column. The task **Synchroniser le dépôt local avec GitHub** is moved to the **Done** column. The second screenshot shows the **Close issue** button in the issue details.

Voilà ce que vous devriez ensuite obtenir dans les **"Issues"** :

The screenshot shows the **Issues** page. The **4 Open** and **1 Closed** counts are shown. The list of issues includes **Ajouter un pied de page**, **Ajouter une barre de navigation**, **Ajouter un favicon**, and **Créer la page d'accueil**.

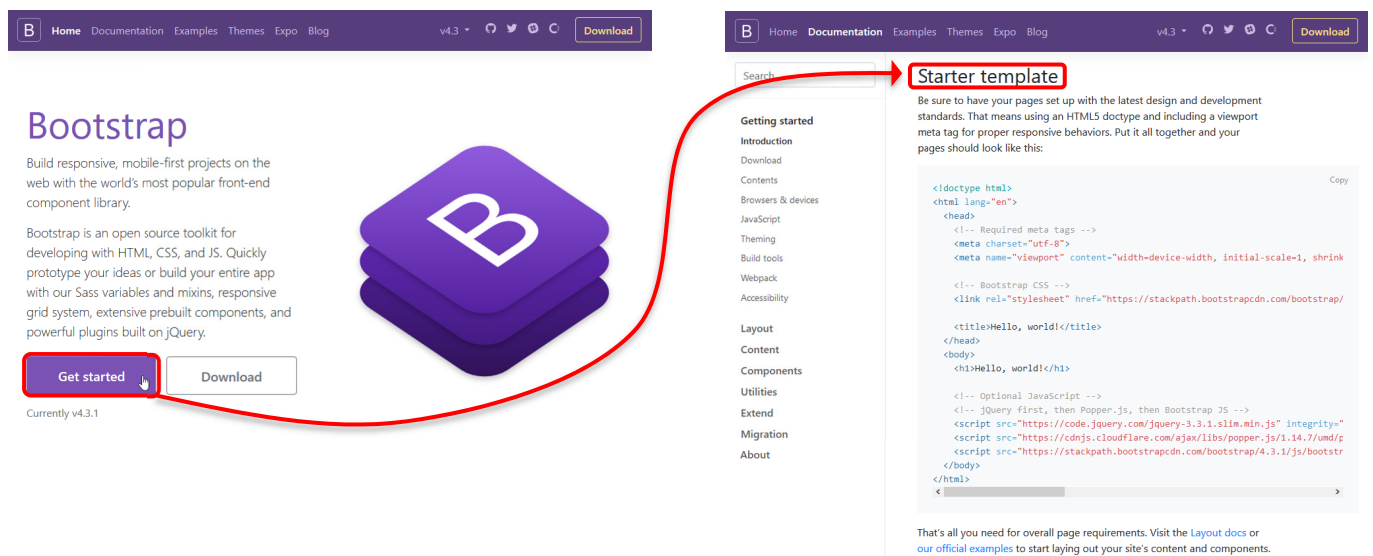
4.4 Création de la page d'accueil

Comme précédemment, pensez à mettre à jour votre Kanban au moment où vous prenez en charge une nouvelle tâche :

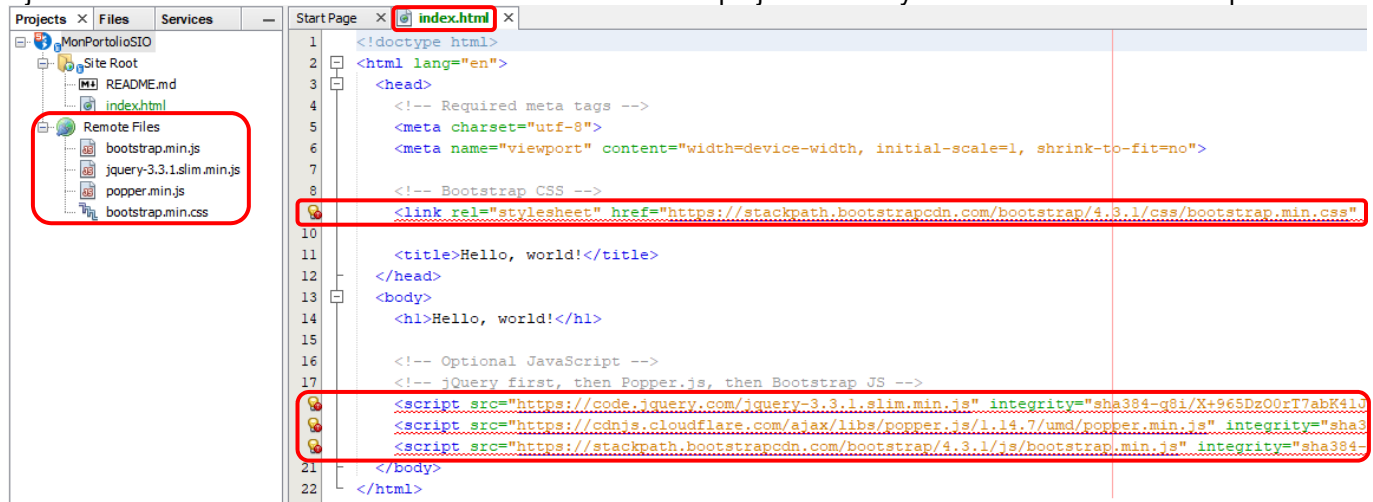


Pour gagner du temps et mettre en œuvre une technologie très largement utilisée sur le web, vous allez utiliser le framework CSS Bootstrap. Pour cela, rendez-vous sur <http://getbootstrap.com> et cliquez sur **"Get started"**.

D'une manière générale, lire la documentation est toujours un bon point de départ. La partie qui nous intéresse ici pour commencer est l'utilisation du **"Starter template"** :



Ajoutez un nouveau fichier HTML "index.html" à la racine du projet et collez-y le contenu du "Starter template" :



Plusieurs choses à relever suite à cette manipulation :

- Le nouveau fichier apparaît en vert. C'est Git qui vous montre que ce fichier ne fait pas partie du dernier "Commit" et n'est donc pas encore inclus dans la gestion de version.
- Dans l'explorateur de projets, une nouvelle rubrique "Remote files" est apparue. C'est parce que dans le contenu que l'on vient d'utiliser, il y a des liens directs vers des fichiers distants.
- NetBeans signale des erreurs car il ne connaît pas l'attribut "integrity". C'est bien un attribut réel mais assez récent. Certains navigateurs ne le supportent que depuis 2018.

TRAVAIL À FAIRE

Pour bien comprendre les lignes utilisées dans le "Starter template" :

- Faites un travail de recherche sur le concept de "CDN", que l'on retrouve dans les noms de serveurs utilisés :

```
<!-- Bootstrap CSS -->
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/

<title>Hello, world!</title>
</head>
<body>
  <h1>Hello, world!</h1>

<!-- Optional JavaScript -->
<!-- jQuery first, then Popper.js, then Bootstrap JS -->
<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" i
<script src="https://cdn.jsdelivr.net/npm/popper.js/1.
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/
```

- Faites un travail de recherche sur l'attribut "integrity" utilisé dans la balise <script>.

Modifiez le code actuel pour l'adapter à notre besoin :

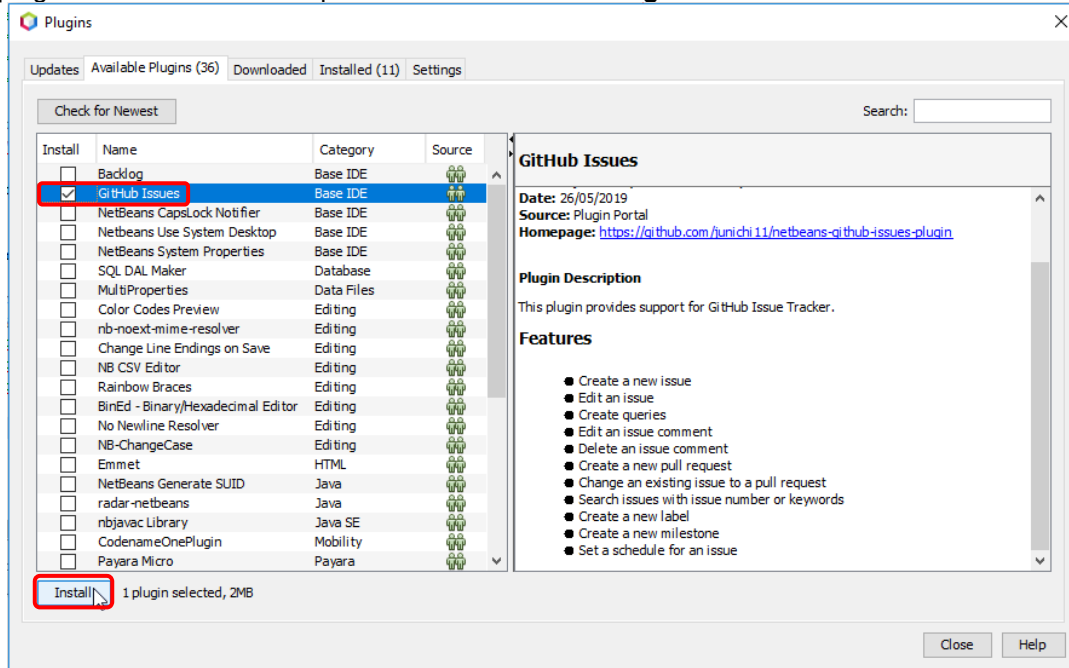
HTML

```
<!doctype html>
<html lang="fr">
  <head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <meta name="description" content="Portfolio d'Amélia Pilou - BTS SIO SLAM">
    <meta name="author" content="Amélia Pilou">
    <!-- Bootstrap CSS -->
    <link rel="stylesheet"
      href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
      integrity="sha384-ggOyR0iXCbMQV3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
      crossorigin="anonymous">
    <title>Portfolio d'Amélia Pilou (BTS SIO 2020)</title>
  </head>
  <body>
    <h1>Bonjour et bienvenu sur mon portfolio !</h1>

    <!-- Optional JavaScript -->
    <!-- jQuery first, then Popper.js, then Bootstrap JS -->
    <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
      integrity="sha384-q8i/X+965Dz00rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
      crossorigin="anonymous"></script>
    <script src="https://cdn.jsdelivr.net/npm/popper.js/1.14.7/umd/popper.min.js"
      integrity="sha384-U02eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1c1HTMga3JDZwnnQq4sF86dIHNDz0W1"
      crossorigin="anonymous"></script>
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"
      integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy60rQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM"
      crossorigin="anonymous"></script>
  </body>
</html>
```

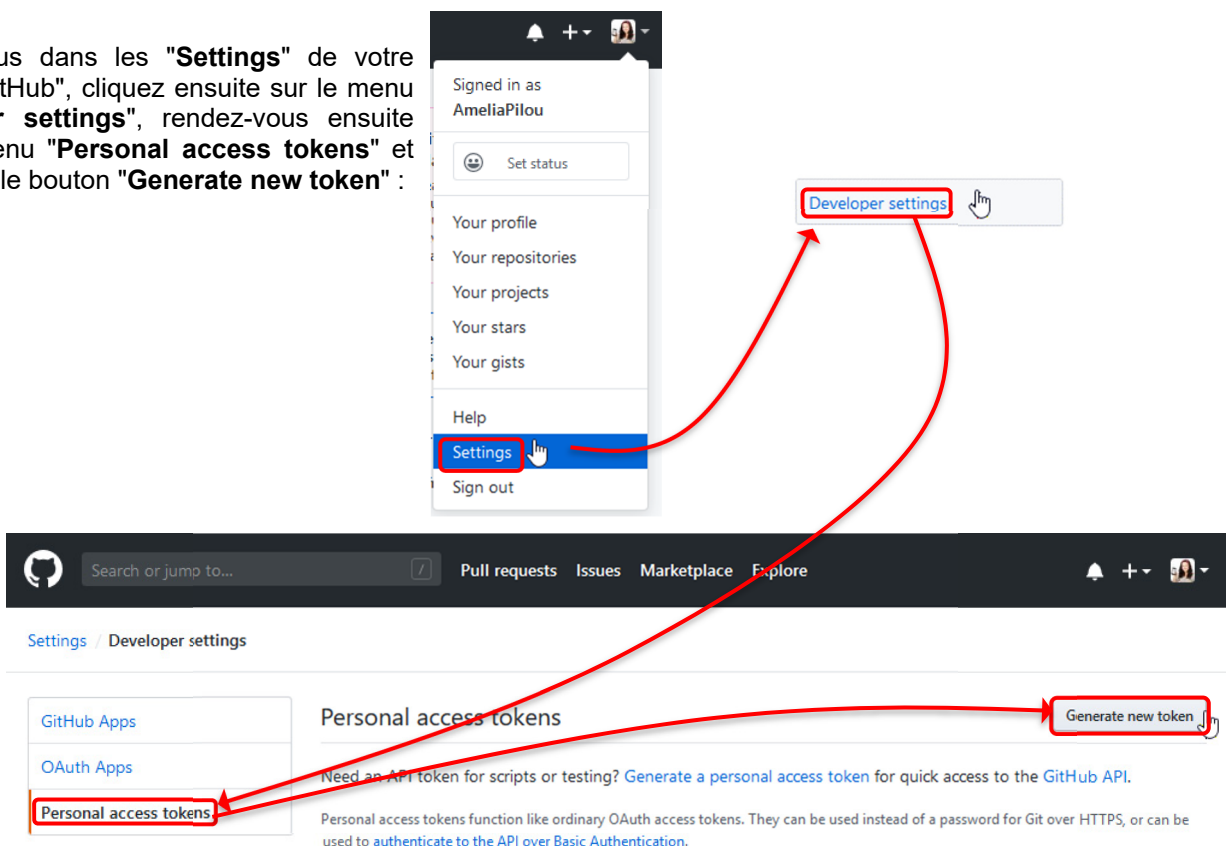
Lier NetBeans à notre système de ticketing GitHub

La page d'accueil est créée. Vous allez donc valider cet état à l'aide d'un **"commit"**. Mais avant de s'occuper du commit à proprement parler, on va lier nos tâches locales avec celles de notre dépôt GitHub. Pour ce faire, installez le plug-in **"GitHub Issues"** depuis le menu **"Tools ► Plugins"**

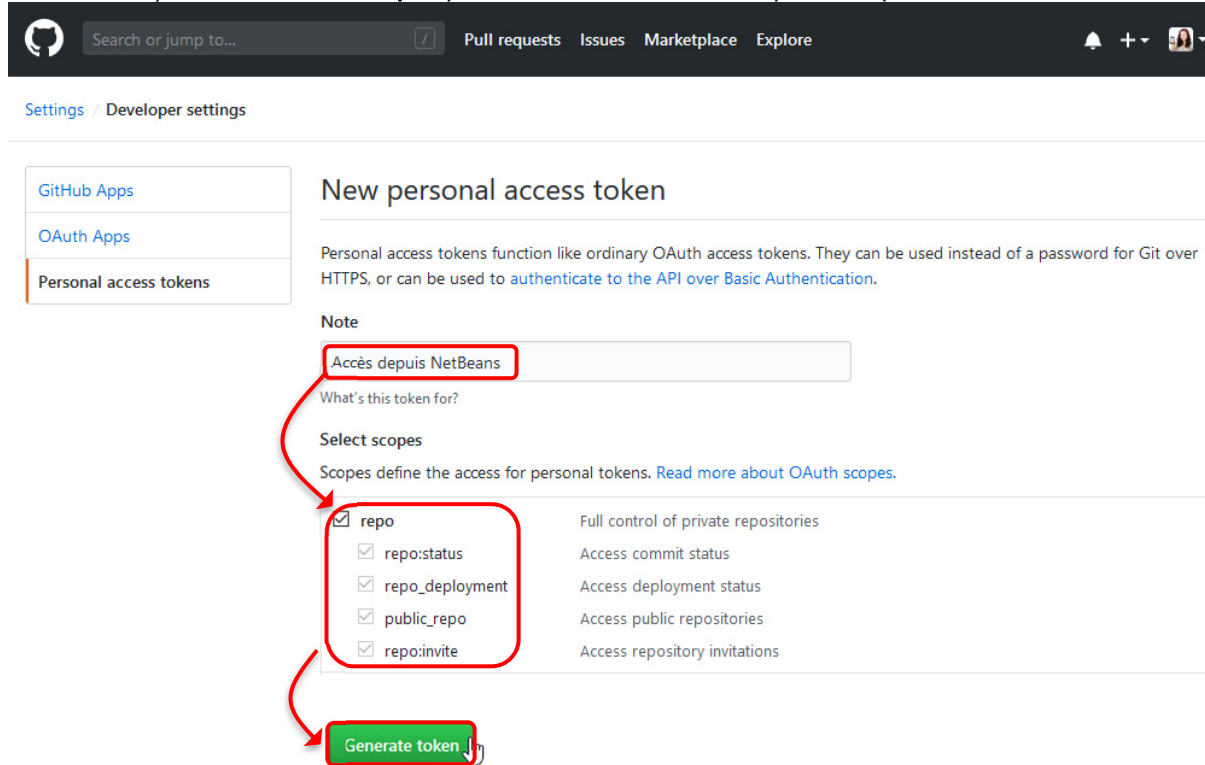


Une fois le plug-in installé, il faut aller récupérer un jeton de connexion sur GitHub, ceci afin que NetBeans puisse se connecter à notre dépôt et plus particulièrement à notre système de gestion d'incidents.

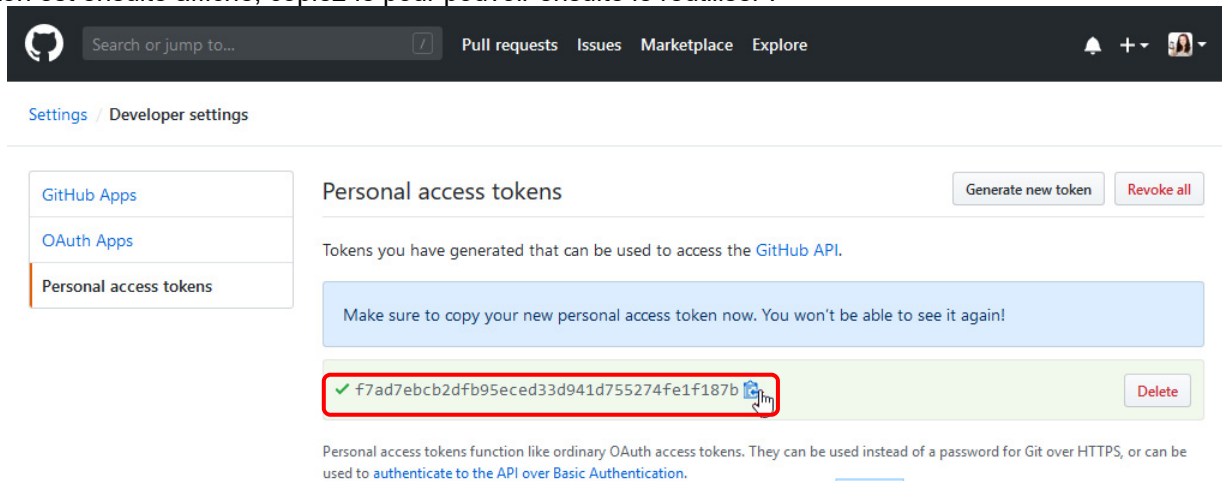
Rendez-vous dans les **"Settings"** de votre compte **"GitHub"**, cliquez ensuite sur le menu **"Developer settings"**, rendez-vous ensuite dans le menu **"Personal access tokens"** et cliquez sur le bouton **"Generate new token"** :



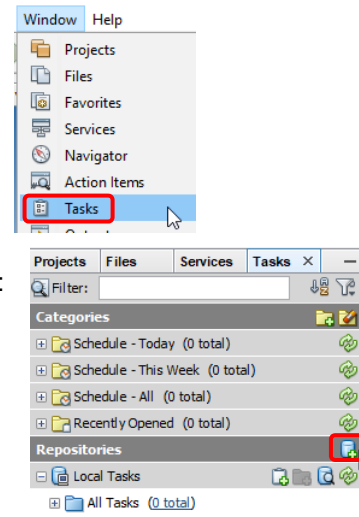
Dans l'écran de personnalisation du jeton, saisissez comme "Note" à quoi va servir ce jeton (par exemple "**Accès depuis NetBeans**"), cochez la case "**repo**" pour autoriser l'accès au dépôt et cliquez enfin sur "**Generate token**" :



Le jeton est ensuite affiché, copiez-le pour pouvoir ensuite le réutiliser :

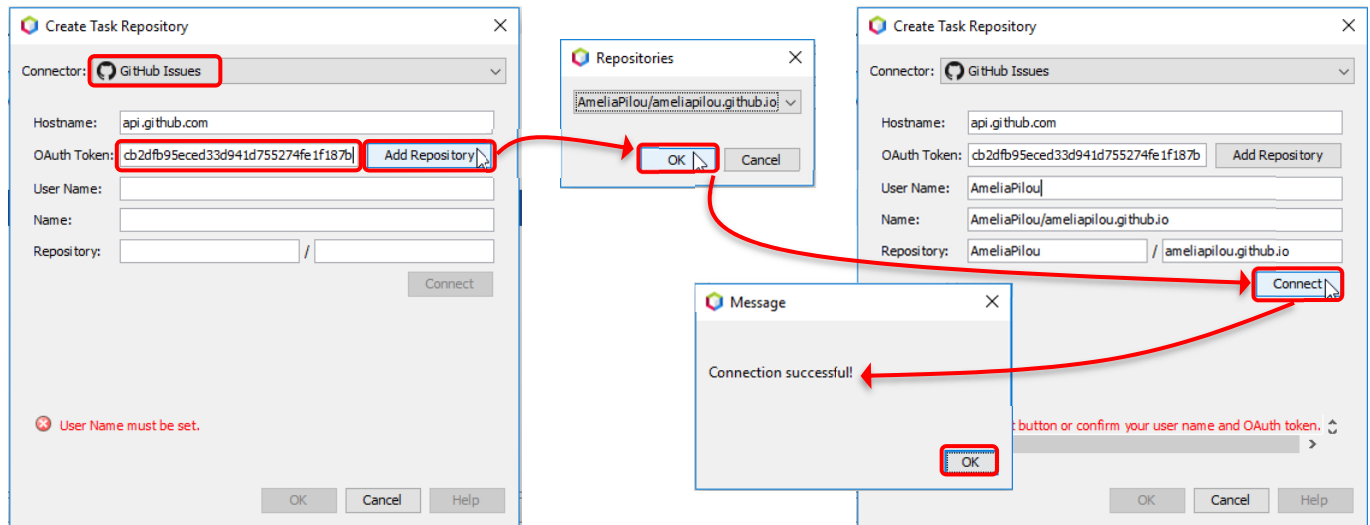


Retournez dans NetBeans, et cliquez sur le menu "**Window ► Tasks**" :

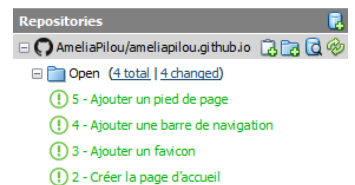


Dans cette nouvelle rubrique, cliquez sur l'icône d'ajout de nouveau dépôt :

Dans la fenêtre d'ajout d'un nouveau dépôt, choisissez comme **"Connector"** **"GitHub Issues"**, l'hostname **"api.github.com"** est normalement déjà présent, collez votre jeton dans **"OAuth Token"** et cliquez sur le bouton **"Add Repository"**. Votre dépôt apparaît normalement dans le menu déroulant, cliquez sur **"OK"**. Les autres zones sont alors automatiquement complétées. Cliquez sur le bouton **"Connect"**, un message vous indiquant le succès apparaît alors, cliquez sur **"OK"** pour valider.

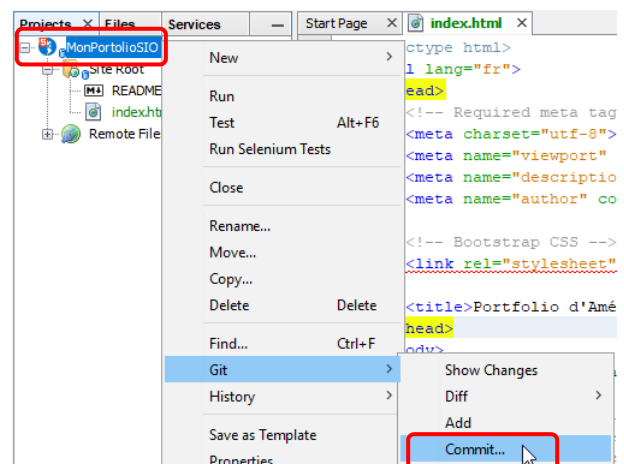


Dans la rubrique **"Tasks"**, vous disposez à présent de votre dépôt et des tickets actuellement "ouverts" :

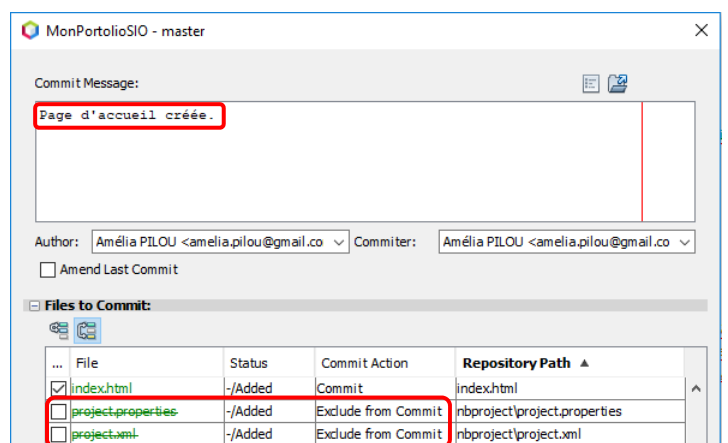


Premier commit

La page d'accueil est créée. On va donc **"entériner"** cette version. Cliquez avec le bouton droit sur le projet puis choisissez **"Git ► Commit..."** (attention à bien utiliser le "Commit" sur le projet, et pas sur le fichier) :



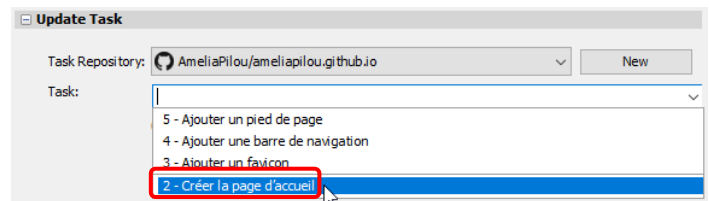
La fenêtre de commit apparaît. Saisissez le **"message"** de commit **"Page d'accueil créée"**, puis décochez les 2 fichiers de configuration de NetBeans (afin de les exclure de la gestion de version) :



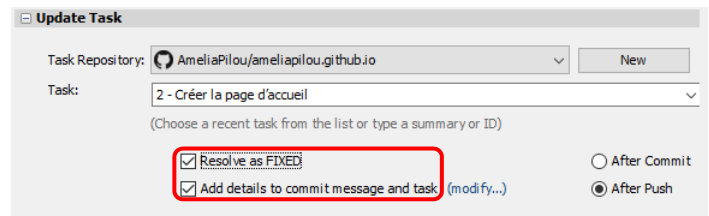
Cliquez ensuite sur la partie **"Update Task"** située dans la partie basse :



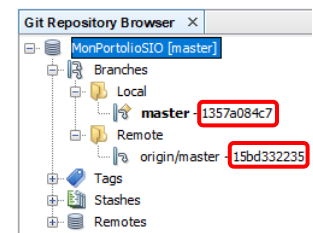
Choisissez ensuite le ticket d'incident correspondant (ici le ticket **"Créer la page d'accueil"**) :



Cochez ensuite les cases **"Resolve as FIXED"** et **"Add details to commit message and task"**, et cliquez sur le bouton **"Commit"** pour valider :

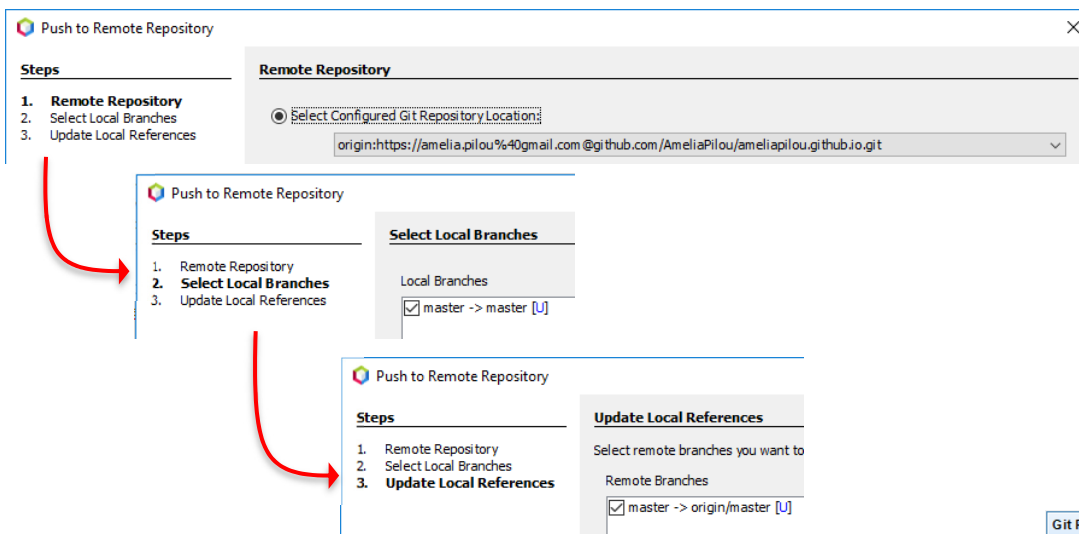


Affichez le **"Git Repository Browser"** en cliquant sur le menu **"Team ► Repository ► Repository Browser"**. On peut voir que la version locale, n'est plus la même que la version distante :

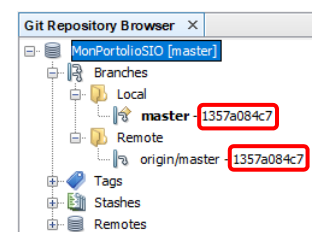


Premier push

On va donc envoyer notre version locale sur le dépôt distant en effectuant un **"Push"**. Pour cela, cliquez avec le bouton droit sur le projet puis choisissez **"Git ► Remote ► Push..."** (attention à bien utiliser le **"Commit"** sur le projet, et pas sur le fichier) :



Après le **"Push"**, on peut alors voir que la version locale est maintenant la même que la version distante et en allant sur votre Kanban en ligne, vous pourrez voir que le ticket a été clos (suite à l'option **"Resolve as FIXED / After Push"** lors du commit) :



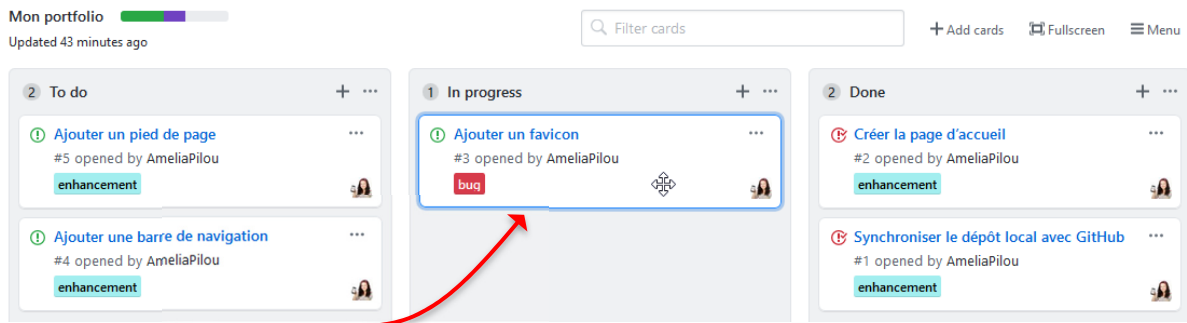
TRAVAIL À FAIRE

Une version est identifiée par une empreinte unique, son **"hash sha1"**. Pour faciliter son utilisation, seuls ses 10 premiers caractères sont affichés. Effectuez des recherches pour comprendre :

- Comment fonctionne une fonction de hachage cryptographique ?
- Mis à part sha1, quels sont les autres algorithmes connus ?

4.5 Ajout d'un favicon

Si vous avez testé la page d'accueil dans un navigateur et que vous êtes allé dans la console, vous avez dû voir une erreur de favicon que l'on va résoudre ici. Comme précédemment, pensez à mettre à jour votre Kanban au moment où vous prenez en charge une nouvelle tâche :



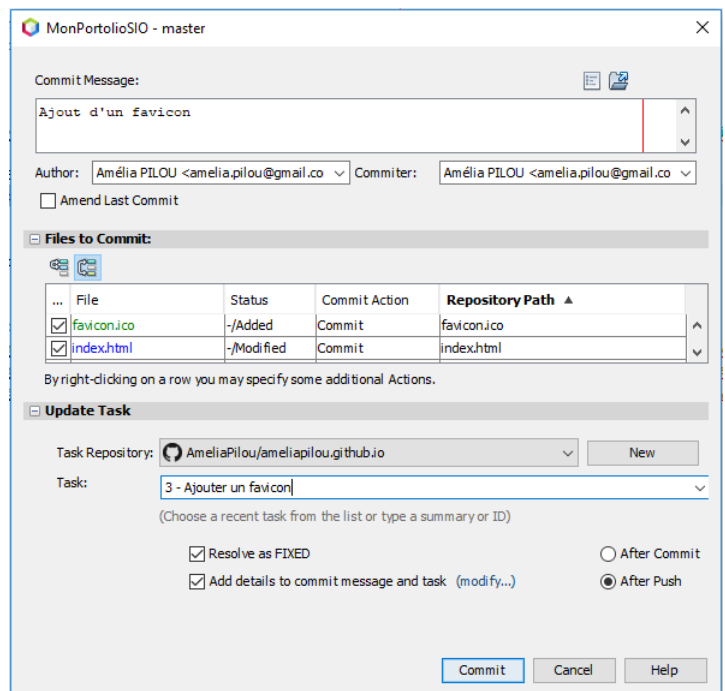
Ajoutez le fichier "favicon.ico" (qui vous a été fourni) à la racine du projet et ajoutez la ligne suivante à la page "index.php" :

HTML

```
<!doctype html>
<html lang="fr">
  <head>
    ...
    <link rel="icon" href="favicon.ico">
  </head>
  <body>
    ...
  </body>
</html>
```

Une fois la modification testée et validée :

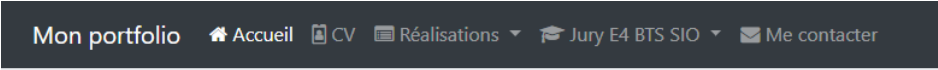
- Faites un "Commit" en n'oubliant pas de saisir un message clair, et en choisissant le ticket à mettre à jour.
- Poussez votre nouvelle version sur GitHub.



4.6 Ajout d'une barre de navigation

Comme précédemment, n'oubliez pas de mettre à jour votre Kanban sur GitHub.

En nous appuyant sur l'aide en ligne (<https://getbootstrap.com/docs/4.3/components/navbar/>) fournie par le framework Bootstrap nous allons créer la barre de navigation suivante :



On va donc déclarer la barre de navigation (navbar), et faire en sorte d'utiliser la grille standard de bootstrap (au travers de la classe "container") :

HTML

```
...
<body>
  <nav class="navbar navbar-dark bg-dark fixed-top navbar-expand-md">
    <div class="container">
      ...
    </div>
  </nav>
</body>
```

On va ensuite créer le nécessaire pour la navigation raccourcie (pour les mobiles) :



Mon portfolio

HTML

```
...
<body>
  <nav class="navbar navbar-dark bg-dark fixed-top navbar-expand-md">
    <div class="container">
      <button type="button" class="navbar-toggler" data-toggle="collapse"
        data-target="#navbar" aria-controls="navbar">
        <span class="sr-only">Navigation raccourcie</span>
      </button>
      <a class="navbar-brand" href="#">Mon portfolio</a>
    </div>
  </nav>
```

On va ensuite déclarer la zone contenant les items de cette barre de navigation :



Mon portfolio

Accueil

CV

Réalisations

Jury E4 BTS SIO

Me contacter

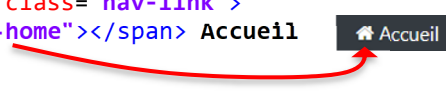
HTML

```
...
<body>
  ...
  <a class="navbar-brand" href="#">Mon portfolio</a>
  <div id="navbar" class="navbar-collapse collapse">
    <ul class="nav navbar-nav">
      ...
    </ul>
  </div>
```

On va ajouter les items de cette barre de navigation en commençant par l'accueil (onglet donc "actif" sur cette page). On notera que pour pouvoir faire apparaître des icônes (la petite maison devant "Accueil") il nous faudra ajouter un lien vers la police d'écriture "**font-awesome**" (on affiche une image à partir d'une classe CSS sur un élément) :

HTML

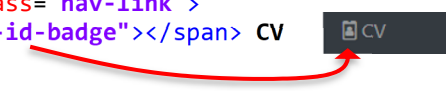
```
<!doctype html>
<html lang="fr">
  <head>
    ...
    .. <link href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css"
        rel="stylesheet">
  </head>
  <body>
    ...
    <ul class="nav navbar-nav">
      <li class="active nav-item">
        <a href="index.html" class="nav-link">
          <span class="fa fa-home"></span> Accueil
        </a>
      </li>
      ...
    </ul>
  </body>
```



De la même façon, on ajoute ensuite l'item "CV" :

HTML

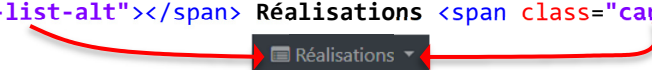
```
...
<body>
  ...
  <li class="nav-item">
    <a href="cv.html" class="nav-link">
      <span class="fa fa-id-badge"></span> CV
    </a>
  </li>
  ...
</body>
```



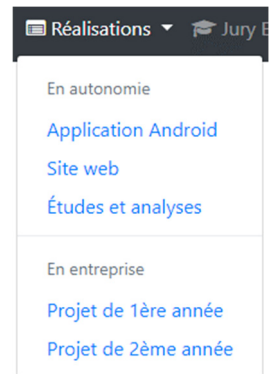
Pour l'item suivant, nous allons utiliser un menu déroulant :

HTML

```
...
<body>
  ...
  <li class="dropdown nav-item">
    <a href="#" class="dropdown-toggle nav-link"
      data-toggle="dropdown" role="button">
      <span class="fa fa-list-alt"></span> Réalisations <span class="caret"></span>
    </a>
  </li>
  ...
</body>
```



Ce menu déroulant sera divisé en deux parties ("En autonomie" et "En entreprise") séparées par une barre de séparation ("divider") et contiendra les items suivants :



HTML

```
...
<body>
...
<li class="dropdown nav-item">
  <a href="#" class="dropdown-toggle nav-link"
    data-toggle="dropdown" role="button">
    <span class="fa fa-list-alt"></span> Réalisations <span class="caret"></span>
  </a>
  <ul class="dropdown-menu" role="menu">
    <li role="presentation" class="dropdown-header dropdown-item">
      En autonomie</li>
    <li role="presentation" class="dropdown-item">
      <a role="menuitem" href="#">Application Android</a>
    </li>
    <li role="presentation" class="dropdown-item">
      <a role="menuitem" href="#">Site web</a>
    </li>
    <li role="presentation" class="dropdown-item">
      <a role="menuitem" href="#">Études et analyses</a>
    </li>
    <li role="presentation" class="dropdown-divider"></li>
    <li role="presentation" class="dropdown-header dropdown-item">
      En entreprise</li>
    <li role="presentation" class="dropdown-item">
      <a role="menuitem" href="#">Projet de 1ère année</a>
    </li>
    <li role="presentation" class="dropdown-item">
      <a role="menuitem" href="#">Projet de 2ème année</a>
    </li>
  </ul>
</li>
...
</body>
```

De la même façon, nous allons ajouter un menu "Jury E4 BTS SIO" qui contiendra tout ce qu'il faut pour pouvoir passer des épreuves dans de bonnes conditions :

Ce menu est volontairement détaillé pour l'exemple, mais il est évident que pour votre portfolio "réel", privilégiez la simplicité avec par exemple 2 pages (SP1 et SP2) où chacune contiendra tout ce qu'il faut pour que le jury puisse travailler sereinement pour vous préparer l'épreuve E4 (documents de contexte, cahier des charges, codes sources, export de bases de données, lien vers les démos en ligne...).

HTML

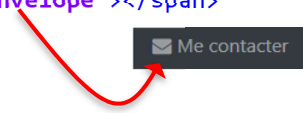
```
...
<body>
...
<li class="dropdown nav-item">
  <a href="#" class="dropdown-toggle nav-link"
    data-toggle="dropdown" role="button">
    <span class="fa fa-graduation-cap"></span>
    Jury E4 BTS SIO
    <span class="caret"></span>
  </a>
  <ul class="dropdown-menu" role="menu">
    <li class="dropdown-header dropdown-item" role="presentation">
      Documentations contexte</li>
    <li role="presentation" class="dropdown-item">
      <a role="menuitem" href="#">Présentation</a></li>
    <li role="presentation" class="dropdown-item">
      <a role="menuitem" href="#">Application</a></li>
    <li role="presentation" class="dropdown-item">
      <a role="menuitem" href="#">Missions à réaliser</a></li>
    <li role="presentation" class="dropdown-item">
      <a role="menuitem" href="#">Annexes</a></li>
    <li role="presentation" class="dropdown-divider"></li>
    <li role="presentation" class="dropdown-header dropdown-item">
      Fiches d'examen</li>
    <li role="presentation" class="dropdown-item">
      <a role="menuitem" href="#">Situation professionnelle n°1</a></li>
    <li role="presentation" class="dropdown-item">
      <a role="menuitem" href="#">Situation professionnelle n°2</a></li>
    <li role="presentation" class="dropdown-divider"></li>
    <li role="presentation" class="dropdown-header dropdown-item">
      Démonstrations</li>
    <li role="presentation" class="dropdown-item">
      <a role="menuitem" href="#">Démon SP n°1</a></li>
    <li role="presentation" class="dropdown-item">
      <a role="menuitem" href="#">Démon SP n°2</a></li>
    <li role="presentation" class="dropdown-divider"></li>
    <li role="presentation" class="dropdown-header dropdown-item">
      Codes sources</li>
    <li role="presentation" class="dropdown-item">
      <a role="menuitem" href="#">Sources SP n°1</a></li>
    <li role="presentation" class="dropdown-item">
      <a role="menuitem" href="#">Sources SP n°2</a></li>
  </ul>
</li>
...
</body>
```

Jury E4 BTS SIO	Me cont
Documentations contexte	
Présentation	
Application	
Missions à réaliser	
Annexes	
Fiches d'examen	
Situation professionnelle n°1	
Situation professionnelle n°2	
Démonstrations	
Démon SP n°1	
Démon SP n°2	
Codes sources	
Sources SP n°1	
Sources SP n°2	

On termine ensuite notre barre de navigation par un menu "Me contacter" :

HTML

```
...  
<body>  
...  
    <li class="nav-item">  
        <a href="#" class="nav-link"><span class="fa fa-envelope"></span>  
        Me contacter  
    </a>  
    </li>  
...  
</body>
```

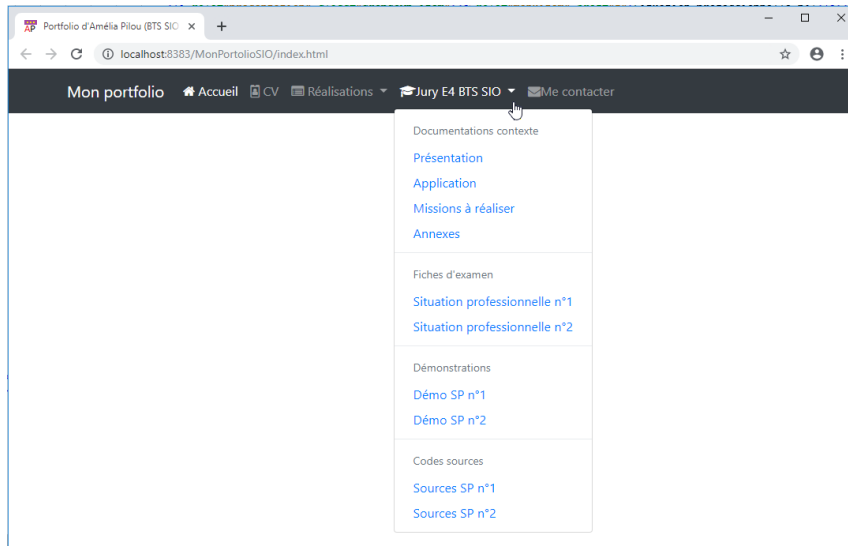


Sans oublier de refermer les éléments HTML ouverts :

HTML

```
...  
<body>  
...  
    </ul>  
    </div>  
    </div>  
    </nav>  
...  
</body>
```

À ce stade, voilà ce que vous devriez obtenir :



Comme précédemment, n'oubliez pas de faire un "Commit" (en précisant que l'on clôt le ticket concerné après le "Push") et un "Push".

Pour que le code soit plus lisible, il ne faut pas hésiter à profiter des fonctionnalités offertes par l'IDE. Par exemple vous pouvez à n'importe quel moment appuyer sur la combinaison **"Alt+Maj+F"** pour que votre code s'indente automatiquement (en cas d'oubli de ce raccourci, il suffit de cliquer sur le menu **"Source"** pour le retrouver à la première place).

4.7 Ajout d'un pied de page

Comme précédemment, n'oubliez pas de mettre à jour votre Kanban sur GitHub.

On va donc déclarer la barre de navigation (navbar), et faire en sorte d'utiliser la grille standard de bootstrap (au travers de la classe "container") :

HTML

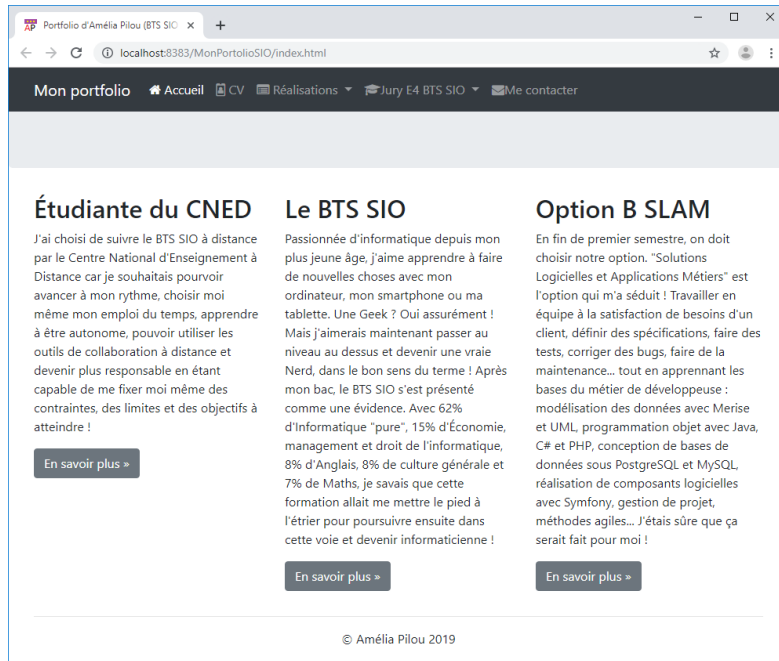
```
...
<body>
  ...
  <div class="jumbotron"></div> <!-- Emplacement pour le contenu principal de la page -->
  <footer>
    <div class="container">
      <div class="row">
        <div class="col-lg-4">
          <h2>Étudiante du CNED</h2>
          <p>J'ai choisi de suivre le BTS SIO à distance par le Centre National
            d'Enseignement à Distance car je souhaitais pourvoir avancer à mon rythme,
            choisir moi-même mon emploi du temps, apprendre à être autonome, pouvoir
            utiliser les outils de collaboration à distance et devenir plus responsable en
            étant capable de me fixer moi-même des contraintes, des limites et des
            objectifs à atteindre !
          </p>
          <p><a class="btn btn-secondary" href="http://www.cned.fr" target="_blank"
            role="button">En savoir plus </a>
          </p>
        </div>
        <div class="col-lg-4">
          <h2>Le BTS SIO</h2>
          <p>Passionnée d'informatique depuis mon plus jeune âge, j'aime apprendre à faire
            de nouvelles choses avec mon ordinateur, mon smartphone ou ma tablette. Une
            Geek ? Oui assurément ! Mais j'aimerais maintenant passer au niveau au-dessus
            et devenir une vraie Nerd, dans le bon sens du terme ! Après mon bac, le BTS
            SIO s'est présenté comme une évidence. Avec 62% d'Informatique "pure", 15%
            d'Économie, management et droit de l'informatique, 8% d'Anglais, 8% de culture
            générale et 7% de Maths, je savais que cette formation allait me mettre le
            pied à l'étrier pour poursuivre ensuite dans cette voie et devenir
            informaticienne !
          </p>
          <p><a class="btn btn-secondary" href="https://fr.wikipedia.org/wiki/Brevet_de_tec
            hnicien_supérieur_-_Services_informatiques_aux_organisations"
            target="_blank" role="button">En savoir plus </a>
          </p>
        </div>
        <div class="col-lg-4">
          <h2>Option B SLAM</h2>
          <p>En fin de premier semestre, on doit choisir notre option. "Solutions
            Logicielles et Applications Métiers" est l'option qui m'a séduit&nbsp;!
            Travailler en équipe à la satisfaction de besoins d'un client, définir des
            spécifications, faire des tests, corriger des bugs, faire de la maintenance...
            tout en apprenant les bases du métier de développeuse : modélisation des
            données avec Merise et UML, programmation objet avec Java, C# et PHP,
            conception de bases de données sous PostgreSQL et MySQL, réalisation de
            composants logiciels avec Symfony, gestion de projet, méthodes agiles...
            J'étais sûre que ça serait fait pour moi !
          </p>
          <p><a class="btn btn-secondary" href="http://www.onisep.fr/Ressources/Univers-For
            mation/Formations/Post-bac/BTS-Services-informatiques-aux-organisations-opt
            ion-B-solutions-logicielles-et-applications-metiers" target="_blank"
            role="button">En savoir plus </a>
          </p>
        </div>
      </div>
    </div>
  </body>
```

```

    </div>
  </div>
  <hr>
  <p class="text-center" id="copyright">© Amélia Pilou 2019</p>
</div>
</footer>    ...
</body>

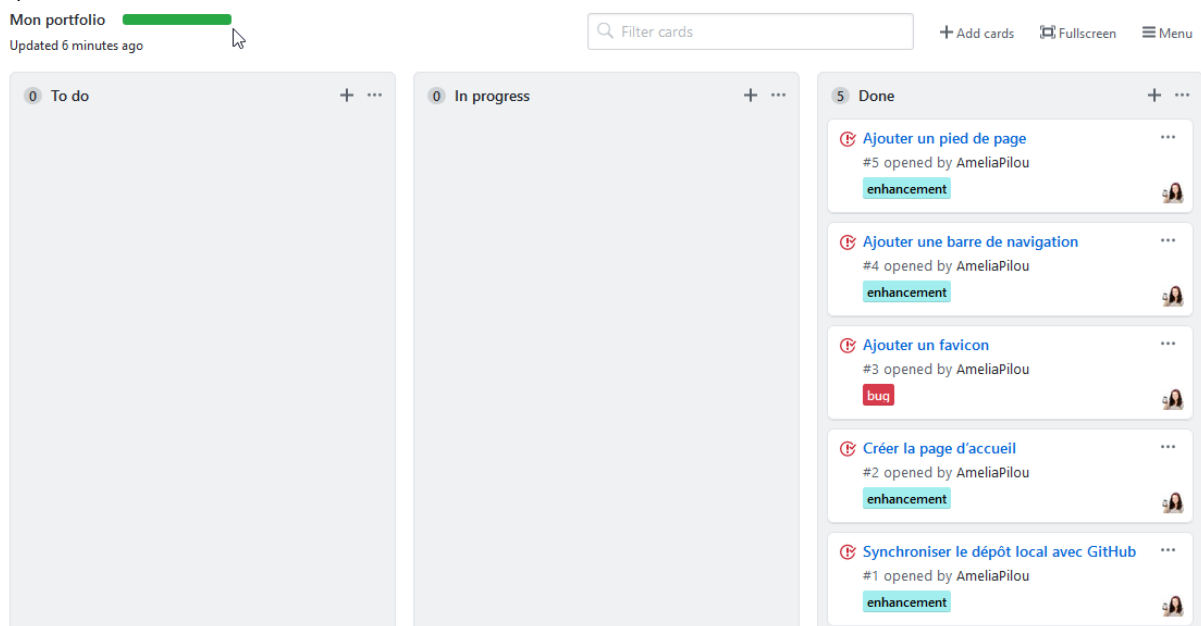
```

À ce stade, voilà ce que vous devriez obtenir :



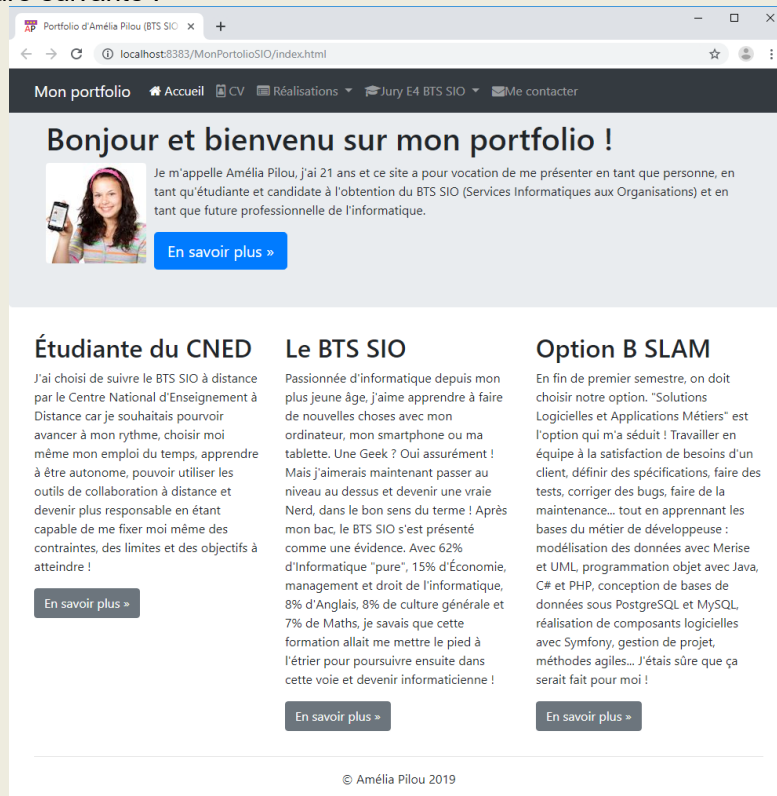
Comme précédemment, n'oubliez pas de faire un "Commit" (en précisant que l'on clôt le ticket concerné après le "Push") et un "Push".

À ce stade, on a terminé la liste de tâches que l'on s'était fixée pour commencer (la barre de progression est pleine et verte) :



TRAVAIL À FAIRE

À l'aide de la photo "AmeliaPilou.jpg" mise à votre disposition, complétez la page d'accueil pour qu'elle ressemble à la capture suivante :



Vous veillerez à conserver les bonnes pratiques d'organisation vues en première année, avec par exemple :

- Placement de la photo dans un répertoire "images" à la racine du projet,
- Si vous avez besoin d'ajouter une feuille de styles, placement elle aussi dans un répertoire "css" à la racine du projet.

Ainsi que les bonnes pratiques d'organisation vues depuis le début de ce TP :

- Ajout d'un ticket sur GitHub avec mise à jour automatique du Kanban
- Commit avec un message clair
- Push sur GitHub avec mise à jour automatique du Kanban

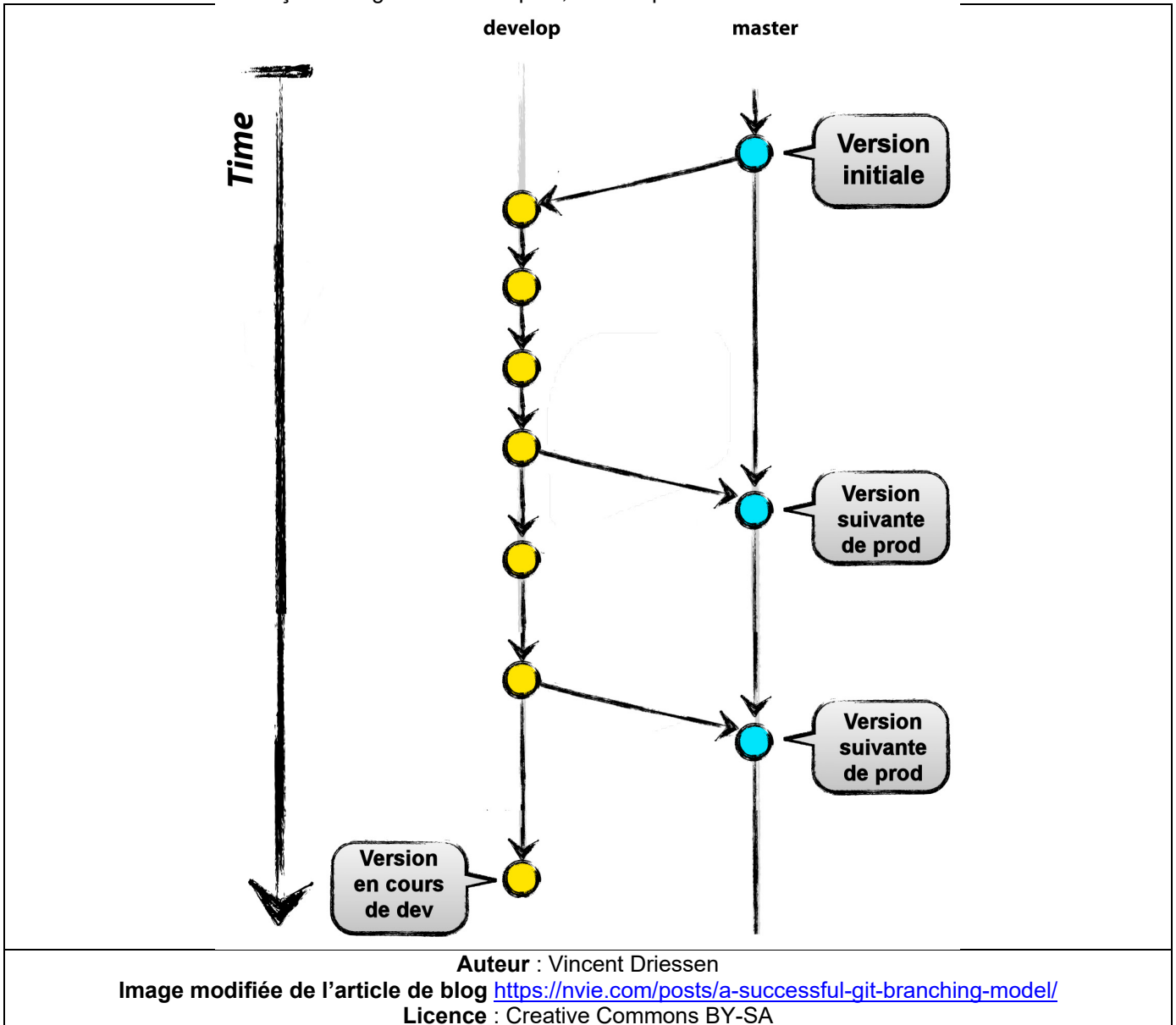
Jusqu'à maintenant, à chaque "Push" nous utilisons la branche par défaut "master". Or, cette branche étant utilisée par les "Pages GitHub", cela signifie qu'à chaque modification de la branche "master", ces modifications étaient directement visibles à l'adresse de nos pages : "<votre nom d'utilisateur>.github.io"

Cela ne respecte pas des principes simples lorsque l'on utilise un système de gestion de version et encore plus lorsque l'on met en œuvre une forge logicielle comme GitHub :

- On ne fait jamais de modifications directement en production ;
- La branche principale doit toujours être stable !

5. Gestion de version avancée

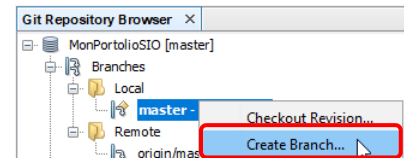
Il existe de nombreuses façons d'organiser ses dépôts, mais la plus courante est la suivante :



5.1 Création d'une nouvelle branche

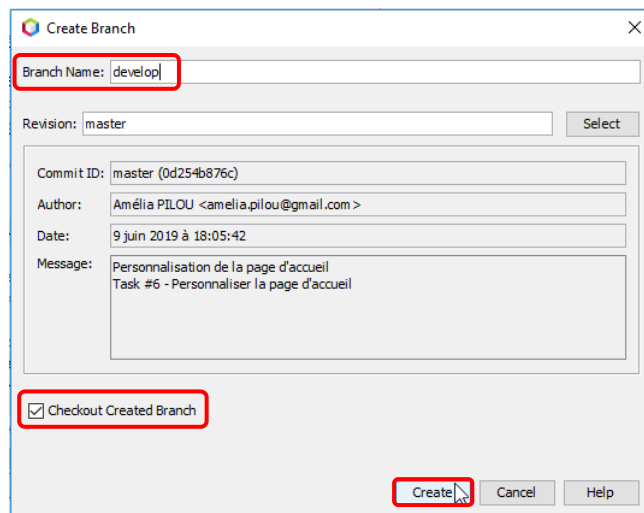
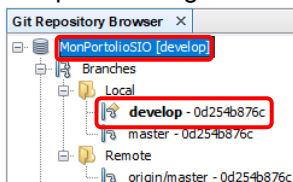
On va commencer par créer la branche "develop", qui par convention est chargée d'accueillir le code en cours de développement :

Dans l'explorateur de dépôt, cliquez avec le bouton droit sur la branche master et choisissez **"Create Branch"** (on peut également utiliser le menu **"Team ► Branch/Tag ► Create Branch"** :



Dans la fenêtre de création de branche, saisissez le nom de la nouvelle branche **"develop"**. On peut voir que le commit à partir duquel on va créer la nouvelle branche est le dernier réalisé (pour récupérer une branche qui sera une copie de l'état actuel de la branche officielle "master"). Cochez la case **"Checkout Created Branch"** pour basculer automatiquement sur la nouvelle branche une fois celle-ci créée et cliquez sur **"Create"** pour lancer la création :

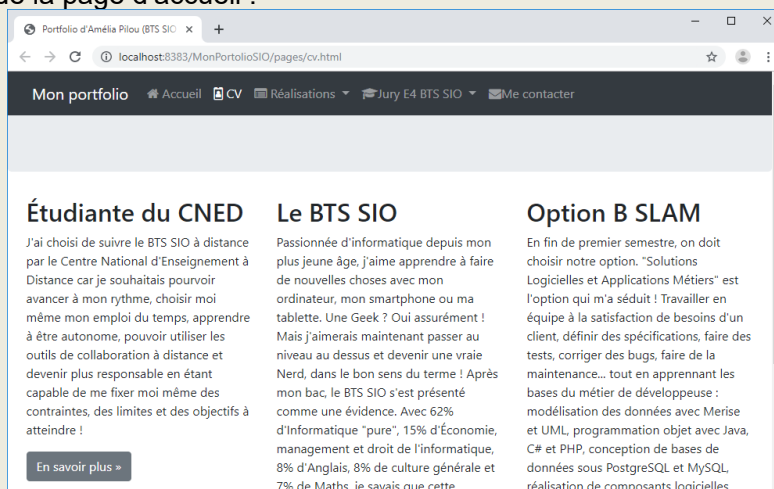
La nouvelle branche apparaît alors dans l'explorateur de dépôt, et l'on peut voir également que l'on a bien basculé dessus :



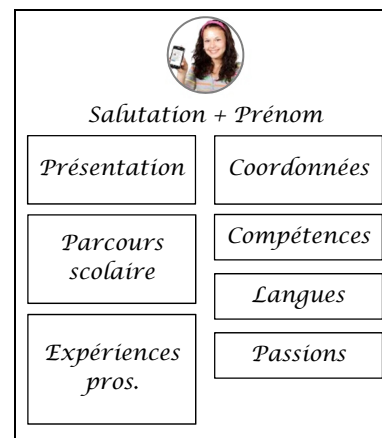
Dès lors, on se situe dans la branche "develop" et toutes les modifications que l'on pourra faire n'auront aucune incidence sur la branche principale (à moins de provoquer une fusion).

TRAVAIL À FAIRE

En vous appuyant sur le projet en cours et sur les bonnes pratiques à mettre en œuvre, vous allez créer une nouvelle page "cv" qui sera placée dans un nouveau répertoire "pages". Cette page devra reprendre la charte graphique de la page d'accueil :



Voilà comment on souhaite voir notre CV (on fera un commit chaque fois que la présentation d'une partie nous satisfera c'est-à-dire à chaque aperçu satisfaisant) :

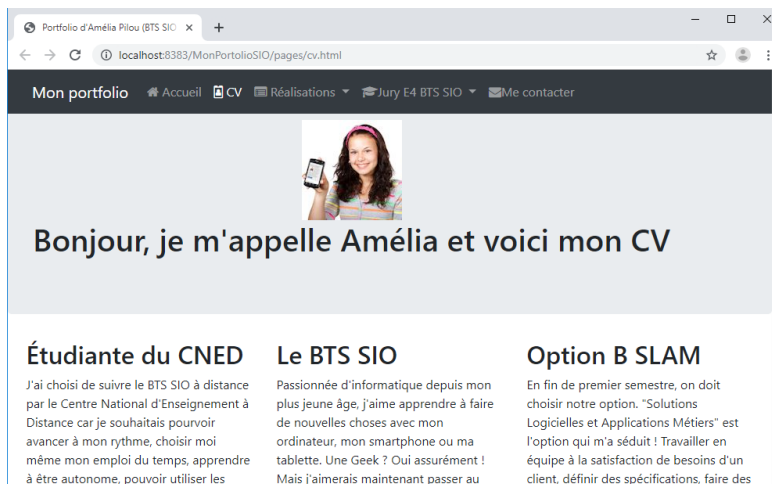


Cette page va avoir des règles de présentations particulières, on va donc créer une nouvelle feuille de styles "cv.css" que l'on va placer dans le répertoire "css" et que l'on va ajouter comme lien dans l'en-tête de la page "cv.html". On va également ajouter un premier contenu :

HTML

```
...
<head>
...
<link href="../css/cv.css" rel="stylesheet">
</head>
<body>
...
<div class="jumbotron">
  <div class="container">
    <div class="row">
      <div class="text-center">
        <div>
          
        </div>
        <div>
          <h1>Bonjour, je m'appelle Amélia et voici mon CV</h1>
        </div>
      </div>
    </div>
  </div>
</div>
...
</body>
</html>
```

Voilà ce que l'on obtient si on n'utilise que les règles Bootstrap :



On va donc créer des identifiants et des classes css pour personnaliser cette première partie :

HTML

```
...
<body>
...
  <div class="row" id="cadreCV">
    <div id="cv-entete" class="text-center">
      <div id="cv-photo">
        
      </div>
      <div id="cv-intro">
        <h1>Bonjour, je m'appelle Amélia et voici mon CV</h1>
      </div>
    </div>
  </body>
</html>
```

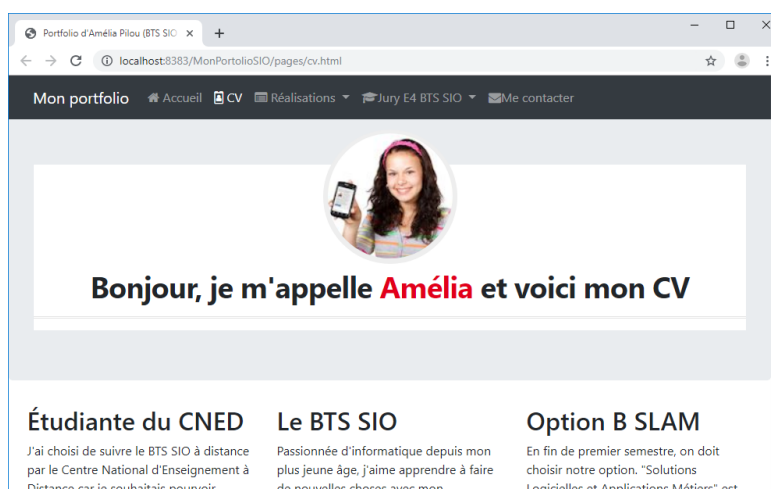
En nous appuyant sur notre code HTML, on va également personnaliser nos règles d'affichage dans le fichier "cv.css" (pour rendre le fichier CSS plus lisible, on va le découper en suivant les découpages du CV à l'aide de ligne de commentaires) :

CSS

```
body { padding-top: 50px; }
.cadreCV { padding-bottom: 15px; background: white; }

/*****
 *      En-tête du CV
 *****/
#cv-entete { margin-top: -45px; width: 100%; }
#cv-photo { width: 172px; height: 172px; border-radius: 50%; overflow: hidden; padding: 5px;
            background: #eee; display: inline-block; }
#cv-photo img { width: 160px; height: 160px; border-radius: 50%; }
#cv-intro h1 { margin: 0; padding: 0; font-size: 2.4em; font-weight: 700;
              letter-spacing: -1px; }
#cv-intro h1 span { color: rgb(226,0,26); }
#cv-intro::after { background: url('images/ligne.png'); /* Dossier "images" est un sous-
                                                         dossier de "css" (à créer) */
                  width: 100%; height: 3px; margin-top: 15px; content: ''; display: block; }
```

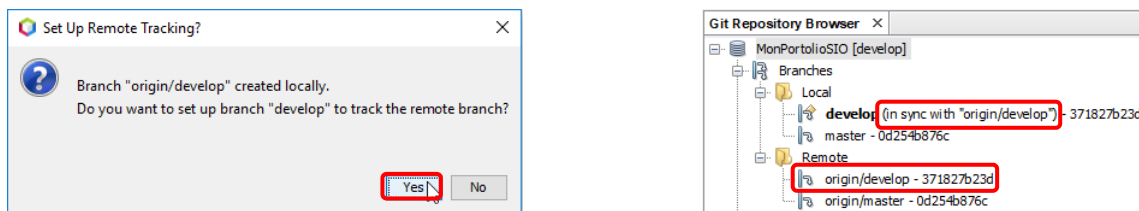
Voilà ce que l'on obtient avec cette personnalisation :



On commit (en actualisant bien la tâche correspondante de notre Kanban que l'on a normalement créé si on respecte nos bonnes pratiques d'organisation). Et ensuite, lorsque l'on va pousser nos modifications sur le dépôt distant, il faudra cette fois-ci que l'on pousse uniquement les modifications faites sur notre branche "develop". Une annotation "A" en vert devrait apparaître à côté de notre branche pour nous indiquer qu'il s'agit d'une nouvelle branche :



Un message apparaît pour nous demander si l'on souhaite suivre les évolutions de la branche distante "develop". Répondre "Yes". Dans l'explorateur de dépôt, on voit alors apparaître :



On ajoute ensuite la deuxième partie du CV elle-même découpée en une partie gauche et une partie droite. On crée donc une nouvelle div de la classe "cadreCV" (pour qu'il y ait une séparation à l'aide du fichier "ligne.png" spécifié dans la règle ci-dessus "#cv-intro:after") et on ajoute une nouvelle div de la classe "col-12" ce qui indique que cette partie devra occuper toute la largeur (12 colonnes sur les 12 de Bootstrap) sur affichage mobile, mais seulement 7 colonnes sur 12 dans les affichages plus grands pour la partie gauche, et 5 colonnes sur 12 pour la partie droite :

HTML

```
...
<body>
...
  <div class="row cadreCV">
    <div class="col-xs-12 col-sm-7">
      <!-- Partie gauche -->
    </div>
    <div class="col-xs-12 col-sm-5">
      <!-- Partie droite -->
    </div>
  </div>
...
</body>
</html>
```

Chaque "rubrique" utilisera des règles communes. On crée donc une classe spécifique dans le fichier "cv.css" :

CSS

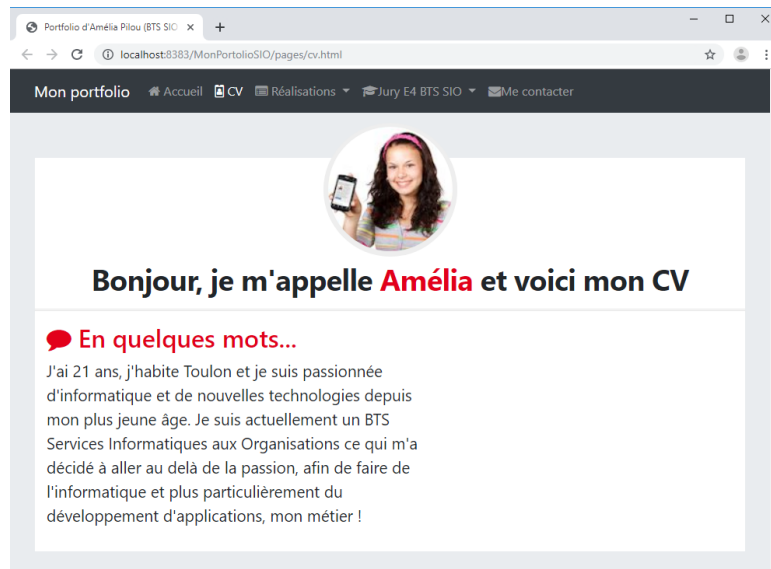
```
...
/*****
*   Rubriques
*****/
.rubrique { margin-bottom: 15px; }
.rubrique h2 { color: rgb(226,0,26); }
.rubrique h2::before { margin-right: 5px; }
.rubrique p { font-size: 1.3em; line-height: 1.5em; }
```

On ajoute donc notre première rubrique de la partie gauche :

HTML

```
...
<body>
...
<!-- Partie Gauche -->
<div class="rubrique clearfix">
  <h2><span class="fa fa-comment"></span> En quelques mots...</h2>
  <p>J'ai 21 ans, j'habite Toulon et je suis passionnée d'informatique et de
    nouvelles technologies depuis mon plus jeune âge. Je suis actuellement un
    BTS Services Informatiques aux Organisations ce qui m'a décidé à aller au
    delà de la passion, afin de faire de l'informatique et plus particulièrement
    du développement d'applications, mon métier&nbsp;nbsp;nbsp;!
  </p>
</div>
...
</body>
</html>
```

Voilà ce que l'on obtient avec cette nouvelle rubrique :



On ajoute ensuite la rubrique "Parcours scolaire" :

HTML

```
...
<body>
...
  <!-- Partie Gauche -->
  ...
  <div class="rubrique clearfix">
    <h2><span class="fa fa-graduation-cap"></span> Parcours scolaire</h2>
    <ul id="parcours" class="clearfix">
      <li>
        <div class="date float-left">
          <span class="fa fa-stop-circle"></span>&nbsp;2020
        </div>
        <div class="details float-right">
          <h3>BTS SIO option SLAM (en cours...)</h3>
          <p>Brevet de Technicien Supérieur "Services Informatiques aux
            Organisations", option "Solutions Logicielles et Applications
            Métiers"</p>
        </div>
      </li>
      <li>
        <div class="date float-left">
          <span class="fa fa-stop-circle"></span>&nbsp;2017
        </div>
        <div class="details float-right">
          <h3>Bac STI2D spécialité SIN</h3>
          <p>Baccalauréat "Sciences et Technologies de l'Industrie et du
            Développement Durable" spécialité "Systèmes d'Information et
            Numérique"</p>
        </div>
      </li>
      <li>
        <div class="date float-left">
          <span class="fa fa-stop-circle"></span>&nbsp;2014
        </div>
        <div class="details float-right">
          <h3>Diplôme Nationale du Brevet</h3>
          <p></p>
        </div>
      </li>
    </ul>
  </div>
...
</body>
</html>
```

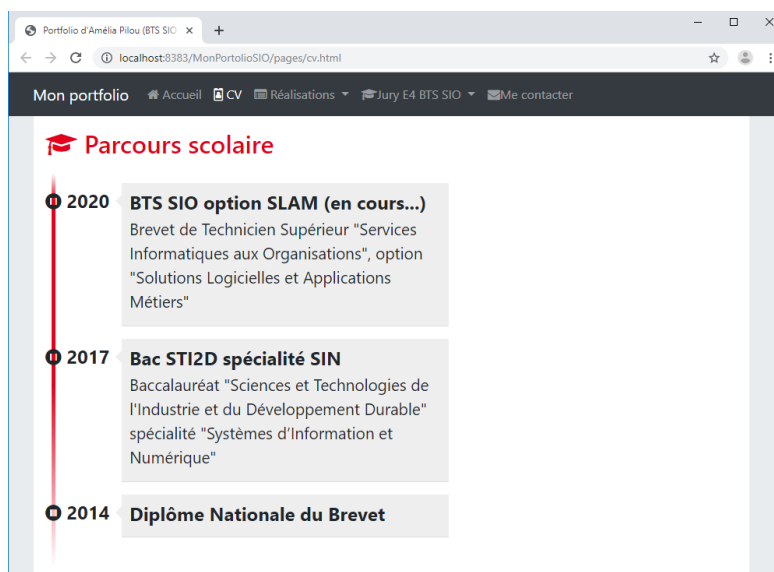
Et on va faire en sorte d'afficher cette rubrique comme une "timeline" en ajoutant les règles CSS suivantes :

CSS

```
...
/*****
*   Parcours scolaire
*****/
#parcours { margin-top: 20px; position: relative; padding: 1em 0; list-style: none; }
#parcours::before { width: 5px; height: 100%; position: absolute; left: 7.8px; top: 0;
                    display: block; background: linear-gradient(to bottom, #ffffff 0%,
                    #E2001A 7%, #E2001A, #ffffff 100%); content: ""; }
#parcours li { width: 100%; z-index: 2; position: relative; float: left; }
#parcours .date .glyphicon-record::before { background: #FFFFFF; color: #555; }
#parcours .date { width: 75px; background-color: rgba(0, 0, 0, 0); font-size: 1.5em;
                  font-weight: 700; display: inline-block; }
#parcours .details { width: 325px; display: inline-block; background: #eee;
                    position: relative; padding: 10px 10px 0 10px;
                    border-bottom: 1px solid #ddd; margin: -4px -12px 20px 0; }
#parcours .details::after { content: ''; position: absolute; top: 15px; right: 0;
                           left: -16px; height: 0; width: 0; border: solid transparent;
                           border-right-color: #eee; border-width: 8px;
                           pointer-events: none; }
#parcours .details h3 { font-size: 1.5em; margin: 0; padding: 0; font-weight: 700; }
#parcours .details p { font-size: 1.3em; margin-top: 5px; padding: 0; }

/* Pour empêcher la déformation du parcours scolaire selon la STA */
@media (min-width: 500px) { #parcours .date { width: 20%; }
                          #parcours .details { width: 80%; }
}
@media (min-width: 768px) { #parcours .date { width: 25%; }
                          #parcours .details { width: 75%; }
}
@media (min-width: 992px) { #parcours .date { width: 17%; }
                          #parcours .details { width: 83%; }
}
@media (min-width: 1200px) { #parcours .date { width: 13%; }
                           #parcours .details { width: 87%; }
}
```

Voilà ce que l'on obtient avec cette nouvelle rubrique :



On ajoute ensuite la rubrique "Expériences professionnelles" :

HTML

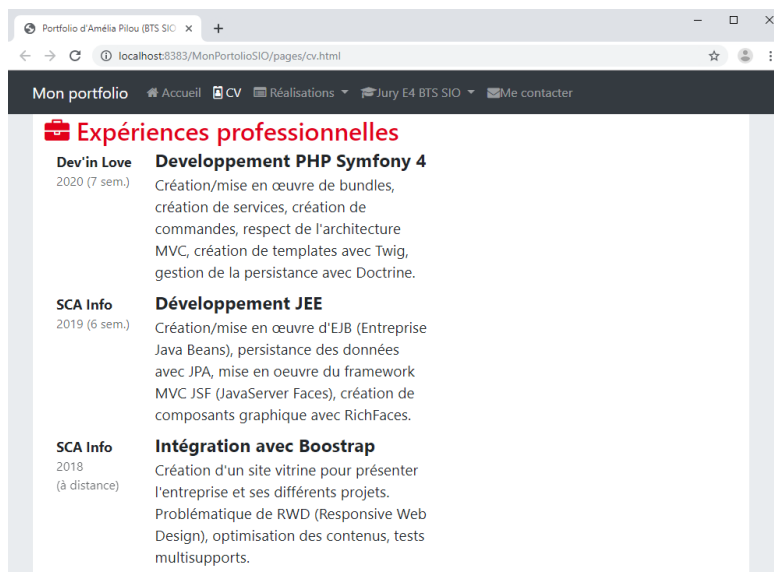
```
...
<body>
...
    <!-- Partie Gauche -->
    ...
    <div class="rubrique clearfix">
        <h2><span class="fa fa-briefcase"></span> Expériences professionnelles</h2>
        <div class="experience clearfix">
            <div class="col-3">
                <div class="lieu">Dev'in Love</div>
                <div class="date">2020 (7 sem.)</div>
            </div>
            <div class="col-9">
                <div class="profession">Developpement PHP Symfony 4</div>
                <div class="details">Création/mise en œuvre de bundles, création de services,
                    création de commandes, respect de l'architecture MVC, création de templates
                    avec Twig, gestion de la persistance avec Doctrine.
                </div>
            </div>
        </div>
        <div class="experience clearfix">
            <div class="col-3">
                <div class="lieu">SCA Info</div>
                <div class="date">2019 (6 sem.)</div>
            </div>
            <div class="col-9">
                <div class="profession">Développement JEE</div>
                <div class="details">Création/mise en œuvre d'EJB (Entreprise Java Beans),
                    persistance des données avec JPA, mise en oeuvre du framework MVC JSF
                    (JavaServer Faces), création de composants graphique avec RichFaces.
                </div>
            </div>
        </div>
        <div class="experience clearfix">
            <div class="col-3">
                <div class="lieu">SCA Info</div>
                <div class="date">2018<br>(à distance)</div>
            </div>
            <div class="col-9">
                <div class="profession">Intégration avec Bootstrap</div>
                <div class="details">Création d'un site vitrine pour présenter l'entreprise et
                    ses différents projets. Problématique de RWD (Responsive Web Design),
                    optimisation des contenus, tests multisupports.
                </div>
            </div>
        </div>
    </div>
    ...
</body>
</html>
```

Et on va faire en sorte de mettre en valeur cette rubrique avec les règles CSS suivantes :

CSS

```
...
/*****
*   Expériences professionnelles
*****/
.experience { display:flex; margin-bottom: 15px; }
.experience .lieu { font-size: 1.1em; font-weight: bold; }
.experience .date { font-size: 1em; opacity: 0.7; }
.experience .profession { font-size: 1.45em; font-weight: bold; margin-top: -6px; }
.experience .details { font-size: 1.2em; line-height: 1.5em; }
```

Voilà ce que l'on obtient avec cette nouvelle rubrique :



5.2 Prise en compte d'un nouveau développeur dans l'équipe

Laissons de côté quelques instants la création du CV, et admettons qu'un nouveau développeur rejoigne le projet. Pour simuler cela, créez un nouveau compte sur GitHub.

Astuce : Si vous disposez d'un compte Gmail, vous disposez également de plusieurs de ses variantes :

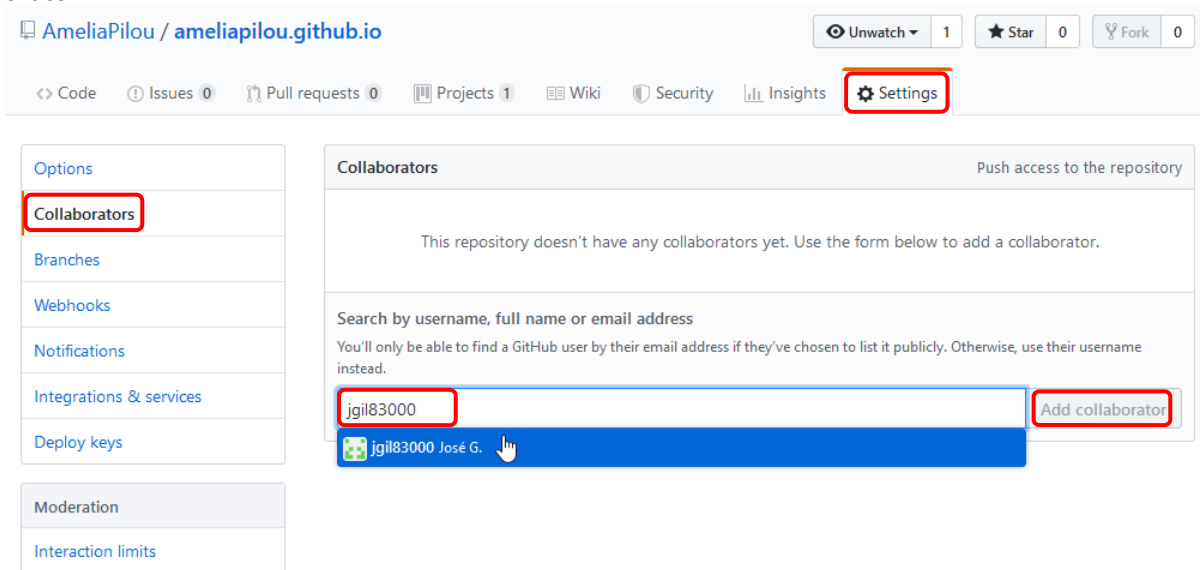
- À base de "points", par exemple si votre email est ameliapilou@gmail.com, vous disposez également de :
 - amelia.pilou@gmail.com
 - a.melia.pilou@gmail.com
 - amelia.pi.lou@gmail.com
 - a.m.eliapilou@gmail.com
 - ...
- Avec un libellé supplémentaire : amelia.pilou+**sio**@gmail.com
- Avec le domaine "googlemail.com" libellé supplémentaire : amelia.pilou@**googlemail**.com

Le mélange reste possible pour créer des combinaisons : a.melia.pilou+**test**@**googlemail**.com

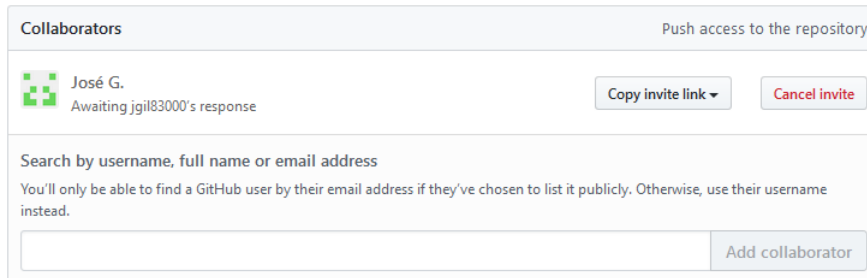
Tous les mails envoyés à ces adresses seront reçus dans la même boîte mail sans avoir à recréer de comptes supplémentaires 😊

Et si vous avez besoin de les utiliser en même temps en ligne (ce que nous allons faire), il vous suffit soit d'ouvrir deux navigateurs différents, soit d'utiliser la navigation privée de votre navigateur habituel.

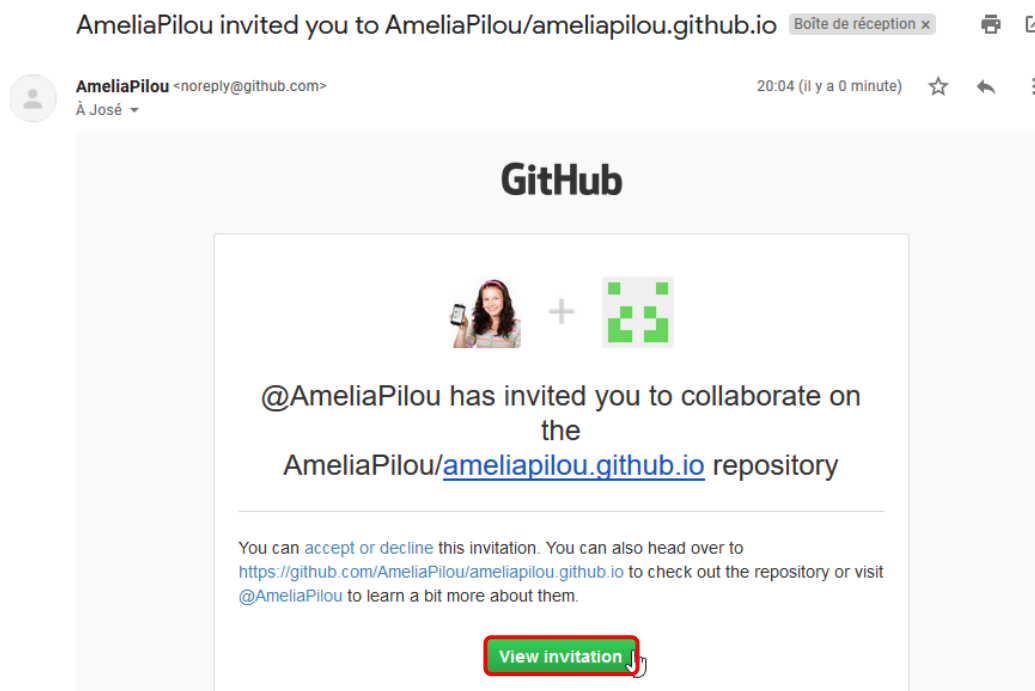
Depuis le premier compte GitHub, rendez-vous dans les **"Settings"** du projet. Allez ensuite dans la partie **"Collaborators"**, déclarez ensuite le compte nouvellement créé et validez en cliquant sur le bouton **"Add collaborator"** :



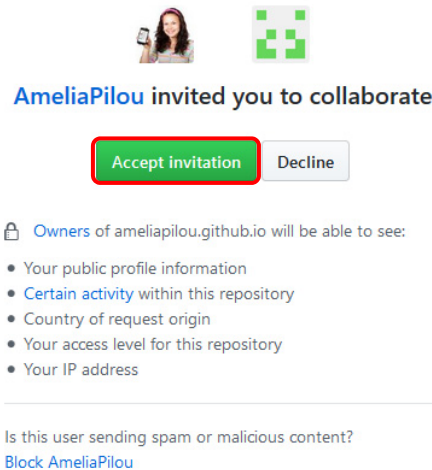
Une fois ajoutée, l'invitation apparaît et peut être retirée à tous moments :



Cette invitation a déclenché l'envoi d'un mail au nouveau collaborateur. Depuis sa boîte, cliquez sur le bouton **"View invitation"** (attention, si vous êtes dans le même navigateur que le premier compte GitHub, copiez ce lien dans un autre navigateur, ou utilisez la navigation privée pour que les deux comptes soient bien distincts) :

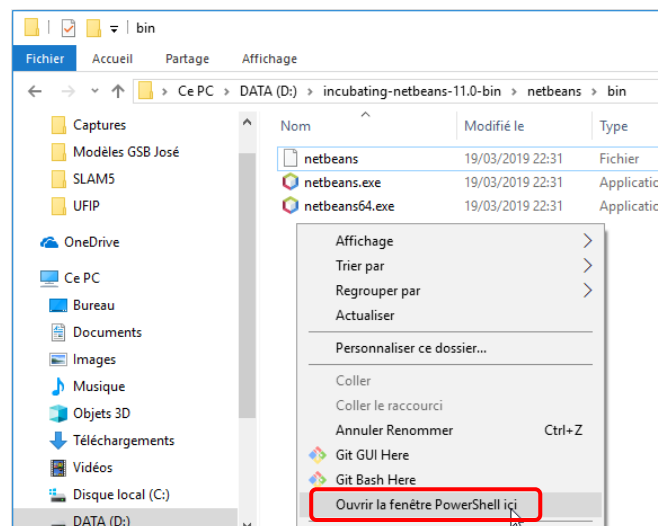


De retour sur GitHub (avec le compte du collaborateur), cliquez sur le bouton **"Accept invitation"** pour valider le fait que le collaborateur rejoint bien le projet initial :

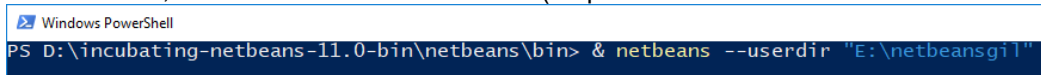


On va maintenant avoir besoin de travailler sur NetBeans en tant que collaborateur. Pour ce faire, il va falloir indiquer à NetBeans que l'on utilise une nouvelle instance de l'IDE avec un nouveau répertoire utilisateur.

Retournez dans le répertoire de lancement de NetBeans, et sans avoir rien de sélectionné, appuyez en même temps sur la touche **"Maj/Shift"** et cliquez avec le bouton droit de la souris sur le fond blanc du dossier, puis choisissez **"Ouvrir la fenêtre PowerShell ici"** (cette option n'apparaît qu'avec cette combinaison de touche) :

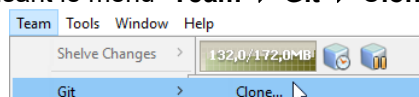


Une fois la fenêtre ouverte, saisissez l'instruction suivante (en personnalisant le chemin de destination) :

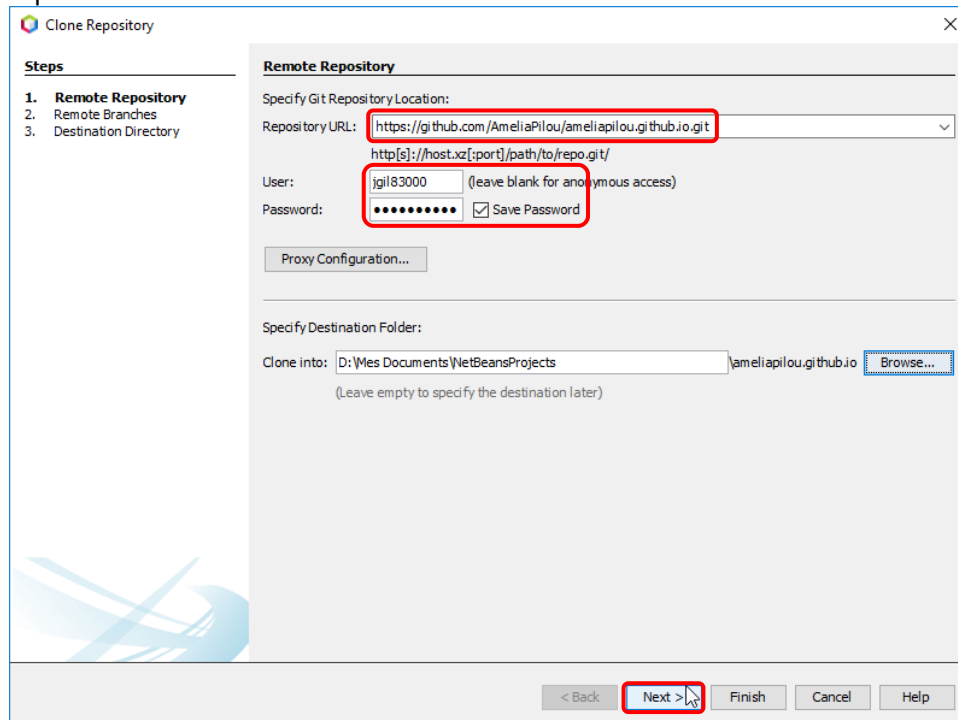


Reconfigurez cette nouvelle instance de NetBeans en ajoutant les plug-ins "HTML5" et "GitHub Issues".

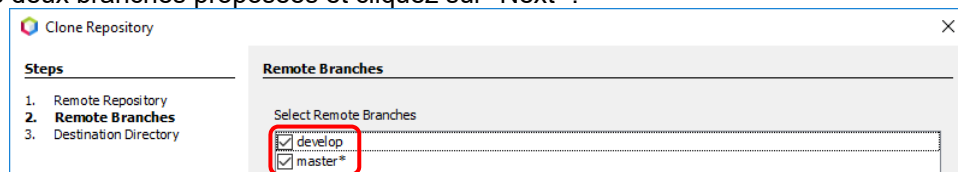
Importer le dépôt depuis GitHub en utilisant le menu **"Team ► Git ► Clone"** :



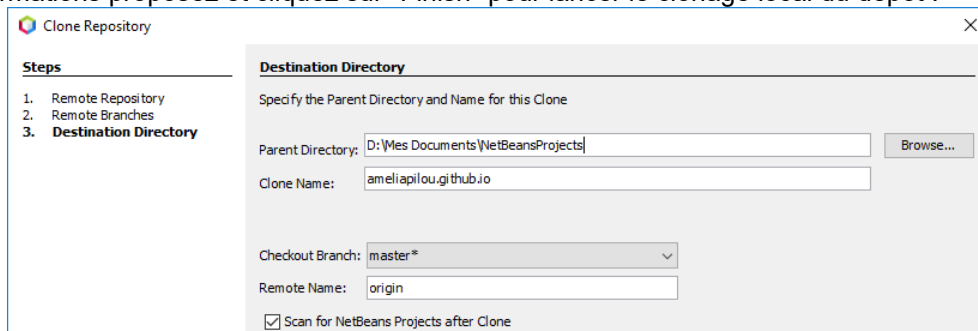
Saisissez les informations de compte du dépôt (à récupérer sur la page d'accueil du dépôt sur GitHub) ainsi que du collaborateur et cliquez sur "Next" :



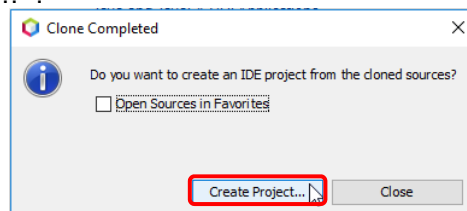
Sélectionnez les deux branches proposées et cliquez sur "Next" :



Validez les informations proposez et cliquez sur "Finish" pour lancer le clonage local du dépôt :

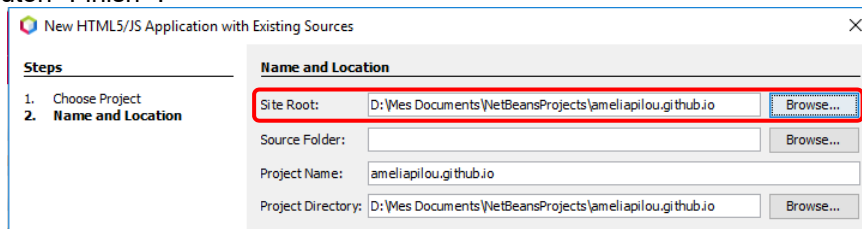


Une fenêtre vous proposant de créer un nouveau projet NetBeans avec le dépôt que vous venez de rapatrier apparaît, cliquez sur "Create Project..." :



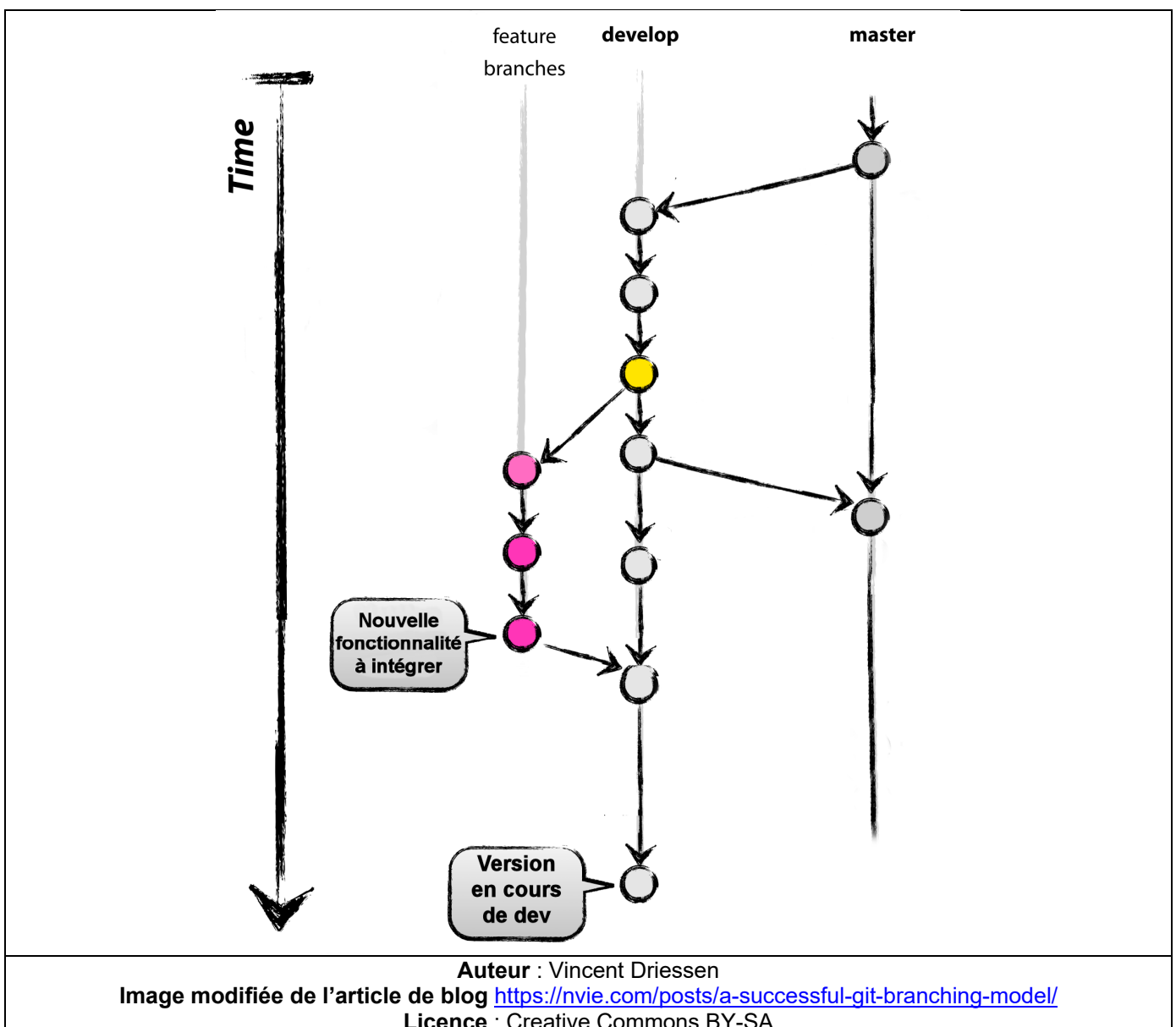
Spécifiez le type de projet "HTML5/JS Application with Existing Sources" et cliquez sur "Next".

Utilisez ensuite l'emplacement où a été rapatrié le dépôt cloné comme "Site Root" et validez la création du projet en cliquant sur le bouton "Finish" :



5.3 Séparation des tâches dans des branches différentes

Seul, ou à plusieurs, il est essentiel de bien s'organiser pour travailler. Dans les bonnes pratiques, il est donc fortement recommandé de subdiviser sa branche de développement en fonctionnalités ("features" en anglais). Par exemple ici, la nouvelle fonctionnalité consiste en l'ajout d'une nouvelle partie "Réalisations" au portfolio qui contiendra plusieurs pages dédiées à chaque réalisation professionnelle effectuée pendant les deux années de BTS : "Application Android", "Site web", "Études et analyses", "Projets réalisés en stage"...



Auteur : Vincent Driessen

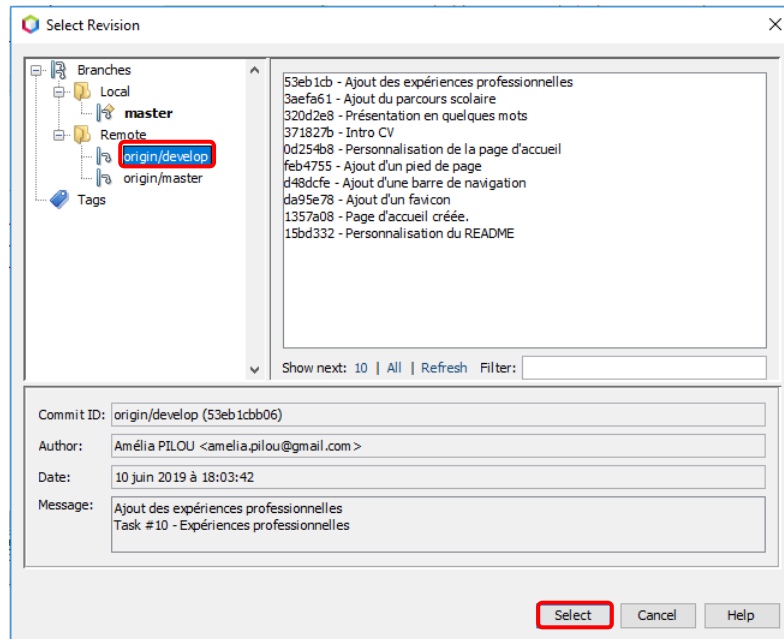
Image modifiée de l'article de blog <https://nvie.com/posts/a-successful-git-branching-model/>

Licence : Creative Commons BY-SA

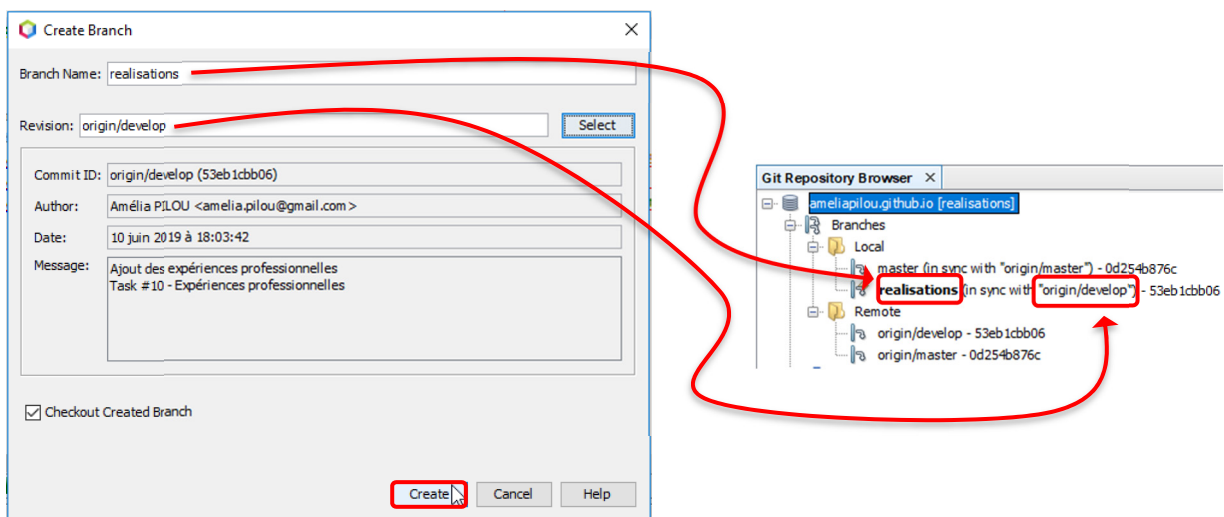
Depuis l'IDE du collaborateur, vous allez donc commencer par créer une nouvelle branche que vous allez appeler "realisations". Cliquez ensuite sur "Select" pour choisir ensuite la branche de laquelle vous souhaitez repartir :



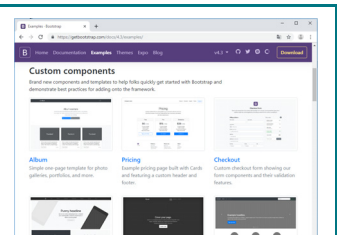
Choisissez ici la branche distante "develop" et cliquez sur "Select" pour valider :



De retour dans la fenêtre de création de branche, cliquez sur "Create" pour lancer la création de la branche "realisations", basée sur la branche distante "develop". Dans l'explorateur de dépôt, vous pouvez retrouver ces informations :



En consultant les exemples de mise en page proposés par Bootstrap (<https://getbootstrap.com/docs/4.3/examples/>) vous allez pouvoir créer rapidement et simplement des visuels pour exposer vos réalisations. Par exemple ici, nous allons nous appuyer sur le premier exemple, l'"Album".



Dans la nouvelle page "appliandroid.html", on va commencer par placer le contenu suivant (créé à partir de l'exemple disponible sur le site de Bootstrap). Dans ce même exemple, des règles CSS sont fournies. Ajoutez donc bien un lien vers "realisations.css" que vous ajouterez au projet :

HTML

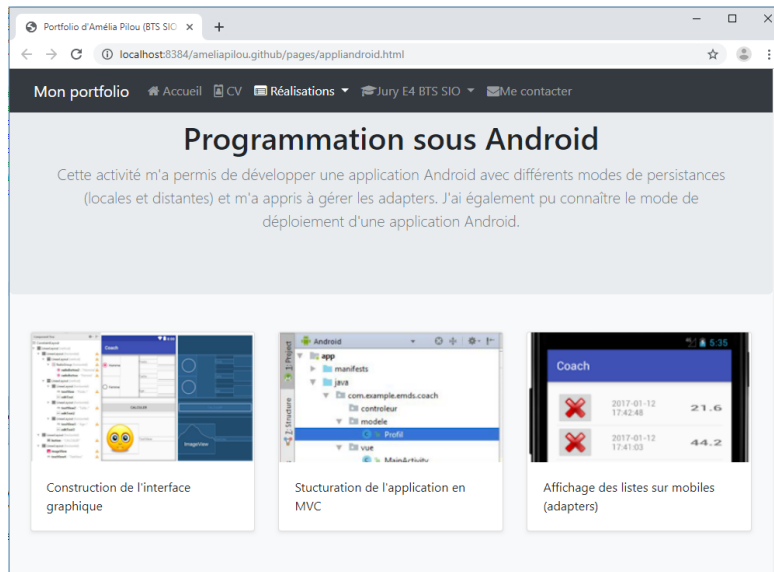
```
...
<head>
  ...
  <link href="../../css/realisations.css" rel="stylesheet">
</head>
<body>
  ...
  <section class="jumbotron text-center">
    <div class="container">
      <h1 class="jumbotron-heading">Programmation sous Android</h1>
      <p class="lead text-muted">Cette activité m'a permis de développer une
        application Android avec différents modes de persistance (locales et
        distantes) et m'a appris à gérer les adapters. J'ai également pu connaître
        le mode de déploiement d'une application Android.
      </p>
    </div>
  </section>
  <div class="album py-5 bg-light">
    <div class="container">
      <div class="row">
        <div class="col-md-4">
          <div class="card mb-4 shadow-sm">
            
            <div class="card-body">
              <p class="card-text">Construction de l'interface graphique</p>
            </div>
          </div>
        </div>
        <div class="col-md-4">
          <div class="card mb-4 shadow-sm">
            
            <div class="card-body">
              <p class="card-text">Stucturation de l'application en MVC</p>
            </div>
          </div>
        </div>
        <div class="col-md-4">
          <div class="card mb-4 shadow-sm">
            
            <div class="card-body">
              <p class="card-text">Affichage des listes sur mobiles (adapters)</p>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
  ...
</body>
</html>
```

Dans le fichier "realisations.css", ajoutez ensuite ce contenu :

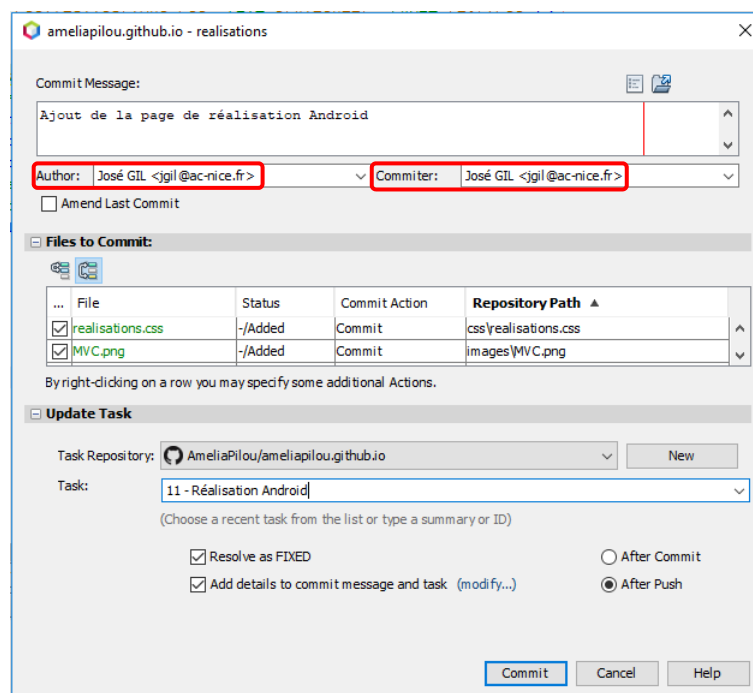
CSS

```
.bd-placeholder-img { font-size: 1.125rem; text-align: middle; -webkit-user-select: none;
                        -moz-user-select: none; -ms-user-select: none; user-select: none; }
@media (min-width: 768px) {
  .bd-placeholder-img-lg { font-size: 3.5rem; }
}
.jumbotron { margin-bottom: 0px; }
```

En ajoutant ensuite les 3 images utilisées au projet, vous devriez obtenir le visuel suivant :



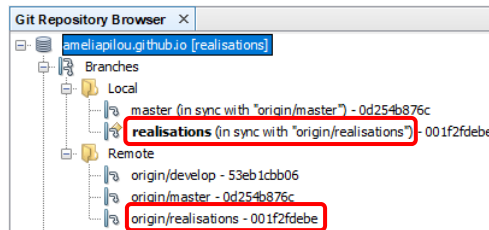
Faites ensuite un commit, en remplaçant bien l'"**Author**" et le "**Committer**" par l'identité du collaborateur au format "**Prénom NOM <email utilisé sur GitHub>**" :



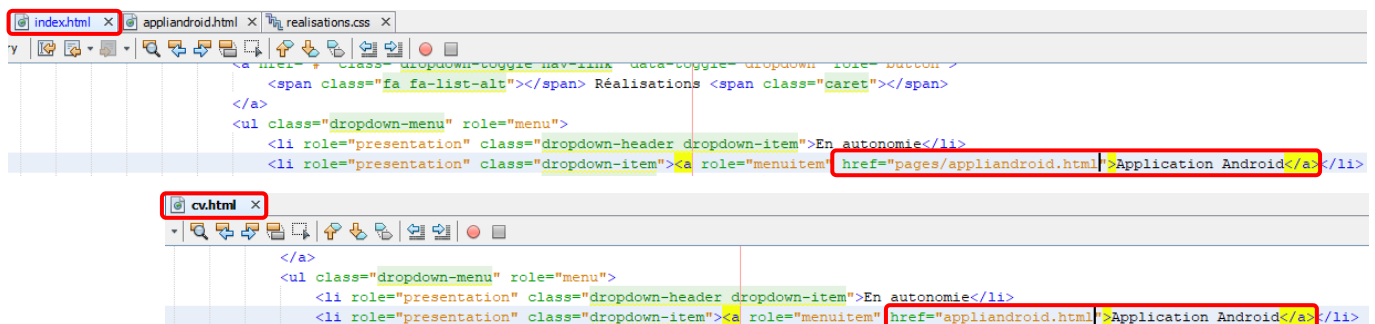
Faites ensuite un "Push". Dans les écrans vous retrouvez bien l'indication de "nouvelle branche" :



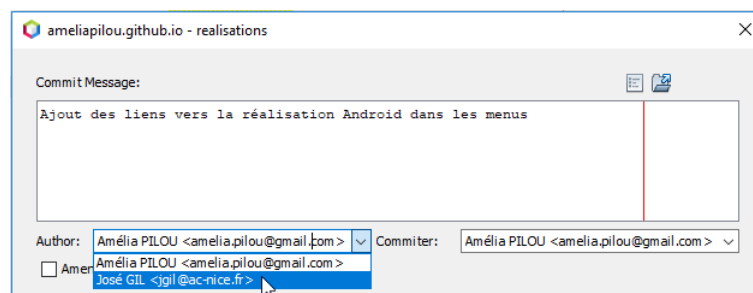
Dans l'explorateur de dépôt, vous pouvez voir que la nouvelle branche distante est bien apparue et que la branche locale "realisations" est maintenant reliée à la branche distante "realisations" :



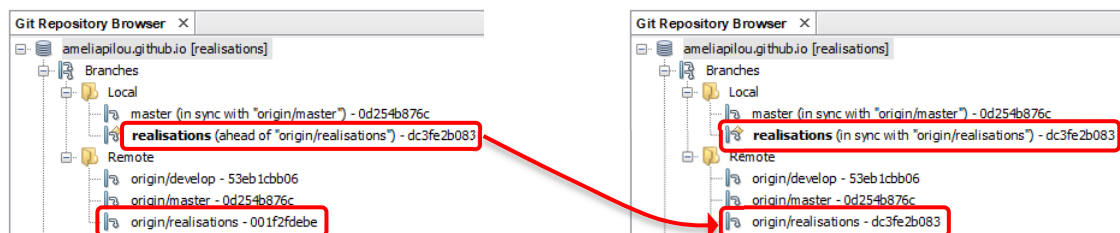
La page étant réalisée, on va mettre à jour les liens dans les pages "index.html" et "cv.html" :



Une fois ces 2 modifications faites, vous pouvez faire un nouveau commit. Cette fois-ci le nom du collaborateur devrait apparaître dans la liste d'"Author" et de "Committer" :

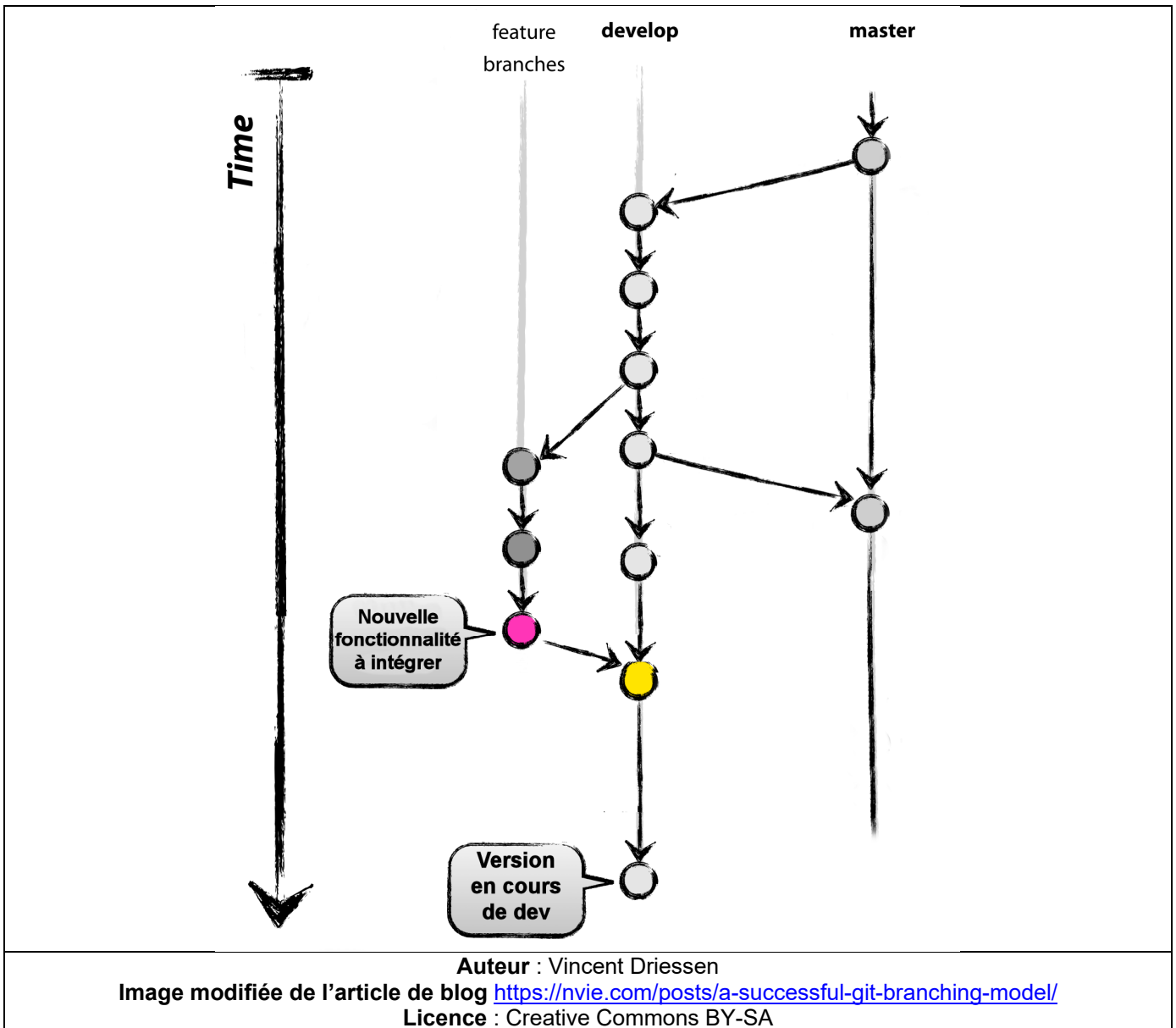


Une fois le commit réalisé, on peut voir que l'identifiant de la branche locale a bien été actualisé. On fait donc un "Push" pour mettre à jour également la branche distante :

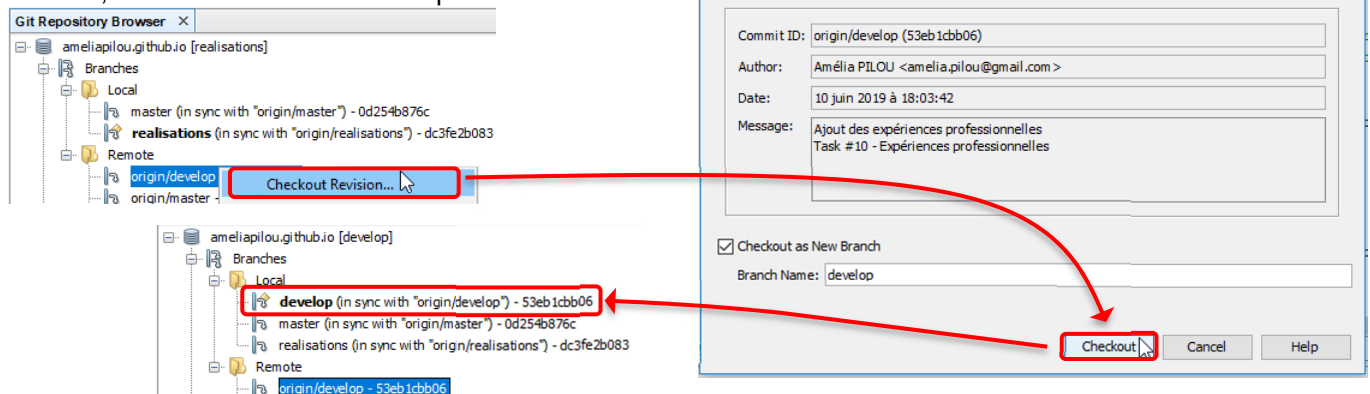


5.4 Fusionner des branches

Admettons que vous ayez terminé cette "fonctionnalité". Vous allez maintenant vouloir fusionner la branche de cette fonctionnalité avec la branche de développement :

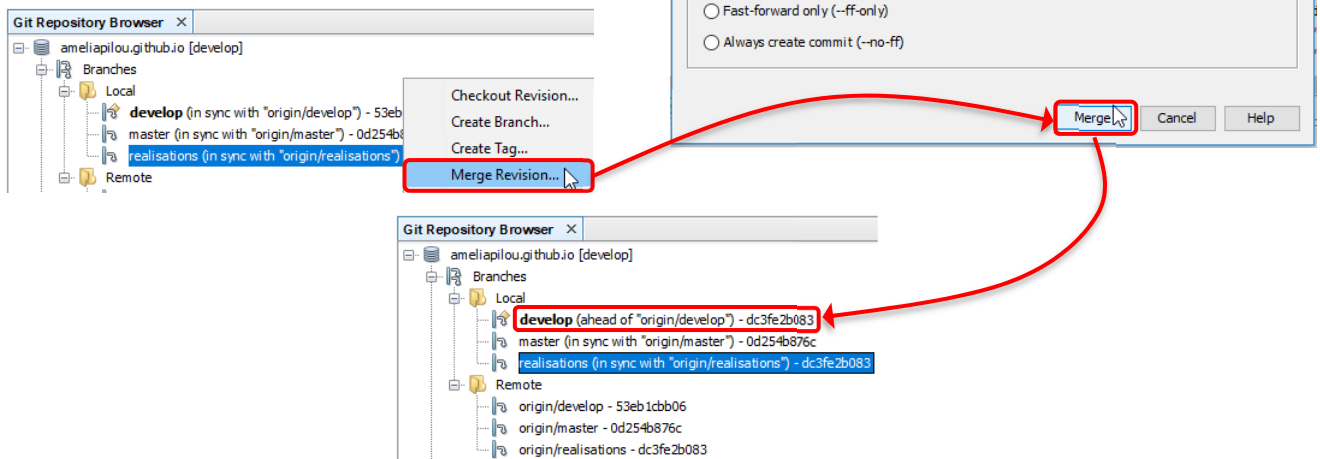


Dans l'explorateur de dépôts, vous allez donc cliquer avec le bouton droit sur la branche distante "**develop**" et choisir l'option "**Checkout Revision**" pour rapatrier en local, la branche distante "develop" :



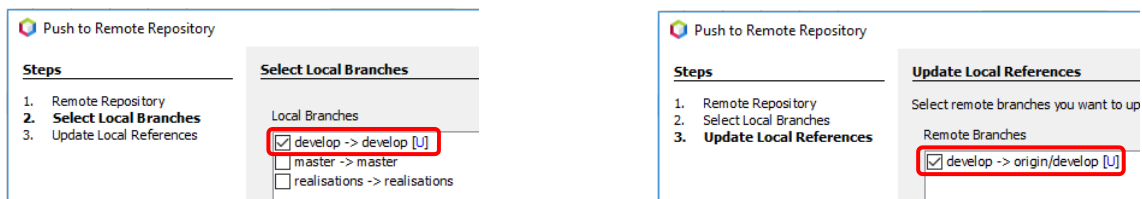
Vous êtes maintenant sur la branche locale "develop". Si vous regardez le code, vous ne devriez plus trouver la page "appliandroid.html" ni toutes les modifications faites sur la branche "realisations", et c'est normal puisque pour l'instant, les modifications faites dans la branche "realisations" n'ont pas été portées sur la branche "develop".

Dans l'explorateur de dépôt, vous allez donc cliquer avec le bouton droit sur la branche locale "realisations" et choisir l'option "Merge Revision..." pour que les 2 branches fusionnent au sein de la branche actuelle :

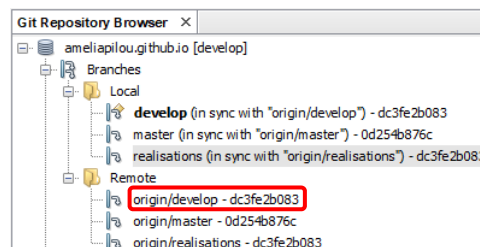


Cette fusion a fonctionné sans incident car la version actuelle de la branche "develop" est celle qui a servi à créer la branche "realisations". Mais si entre-temps il y avait eu de nouvelles versions sur la branche "develop", cette fusion aurait été bloquée par le système de gestion de version.

Faites à présent un "Push" de la branche "develop" pour que le résultat de la fusion soit pris en compte sur la branche distante. Vous pouvez alors voir l'indication de mise à jour à réaliser ("U") de la branche distante :



Vous pouvez ensuite voir que la branche distante est maintenant de la même version que la branche locale :



5.5 Les garde-fous des systèmes de gestion de versions

Retournez sur l'IDE principal (celui d'Amélia dans notre cas) et ajoutez au CV la rubrique "Me contacter" :

HTML

```
...
<body>
...
    <!-- Partie Droite -->
    <div class="rubrique clearfix">
    <h2><span class="fa fa-user"></span> Me contacter</h2>
    <div class="meContacter">
        <div class="details float-left">
            <a href="mailto:amelia.pilou@gmail.com?subject=Contact depuis le CV en ligne"
                title="Mail de Amélia Pilou">
                 GMail.com
            </a>
        </div>
    </div>
    <div class="meContacter">
        <div class="details float-left">
            <a href="https://twitter.com/AmeliaPILOU"
                title="Lien vers la page Twitter de @AmeliaPilou">
                 Twitter
            </a>
        </div>
    </div>
    <div class="meContacter">
        <div class="details float-left">
            <a href="http://fr.linkedin.com/in/ameliaPilou"
                title="Lien vers le profil LinkedIn de Amélia Pilou">
                 LinkedIn
            </a>
        </div>
    </div>
    <div class="meContacter">
        <div class="details float-left">
            <a href="https://github.com/AmeliaPilou/"
                title="Lien vers le profil GitHub d'Amélia Pilou">
                 GitHub
            </a>
        </div>
    </div>
    <div class="meContacter">
        <div class="details float-left">
            <a href="skype:ameliapilou?userinfo"
                title="Lien vers le profil Skype d'Amélia Pilou">
                 Skype
            </a>
        </div>
    </div>
    <div class="meContacter">
        <div class="details float-left">
            <a href="https://instagram.com/AmeliaPilou"
                title="Lien vers le profil Instagram d'Amélia Pilou">
                 Instagram
            </a>
        </div>
    </div>
```

```

        </a>
      </div>
    </div>
  </div>
  ...
</body>
</html>

```

Dans le fichier "cv.css", ajoutez ensuite ce contenu :

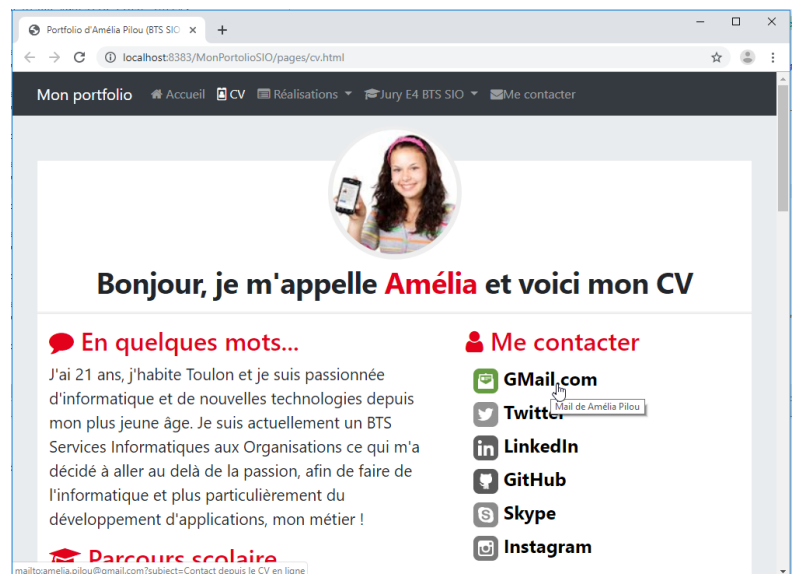
CSS

```

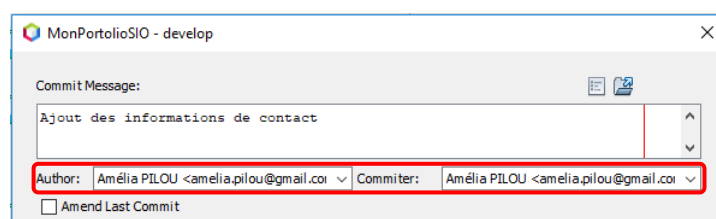
/*****
 *   Me contacter
 *****/
.meContacter { clear: both; }
.meContacter .details { width: 100%; margin-left: 10px; margin-top: 3px; font-size: 1.5em;
                        font-weight: 700; justify-content: space-between; }
.contact-icon { filter: grayscale(1); -webkit-filter: grayscale(1);
                -moz-filter: grayscale(1); -ms-filter: grayscale(1);
                -o-filter: grayscale(1); height: 32px; width: 32px; margin: 2px 0; }
.meContacter a, .meContacter a:hover { text-decoration: none; color: black; }
.meContacter a:hover>img { filter: grayscale(0); -webkit-filter: grayscale(0);
                            -moz-filter: grayscale(0); -ms-filter: grayscale(0);
                            -o-filter: grayscale(0); }

```

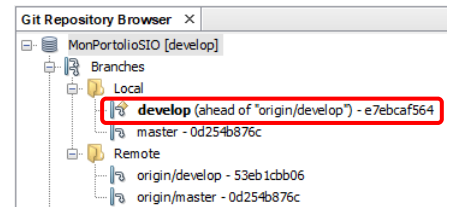
Vous devriez obtenir le visuel suivant :



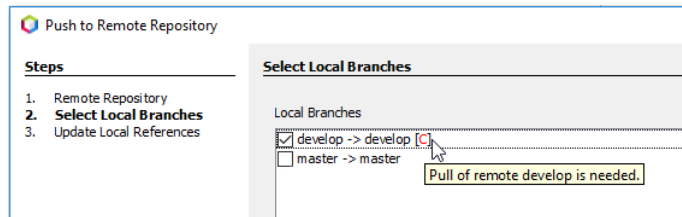
Une fois ces modifications effectuées, faites un nouveau commit. Cette fois-ci avec le compte d'"Author" et de "Committer" original :



Dans l'explorateur de dépôts, on peut voir que la version locale est maintenant différente de la distante.



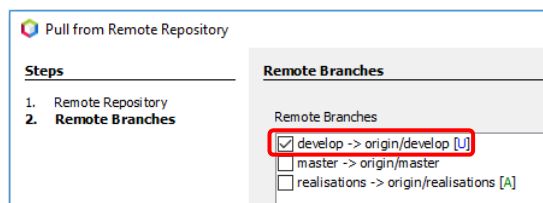
Faites à présent un **"Push"** de la branche **"develop"** :



On peut voir qu'au travers d'une note rouge entre crochets **"C"**, le système de gestion de version vous informe que le push ne sera pas possible (à moins de le **"forcer"** ce qui serait stupide) car depuis la dernière fois que vous avez récupéré cette branche, il y a eu des modifications (celles faites par le collaborateur). En passant la souris dessus, la solution à ce problème s'affiche : Il faut d'abord faire un **"Pull"** pour pouvoir ensuite faire un **"Push"** ! Cliquez sur **"Cancel"** pour annuler la demande de push.

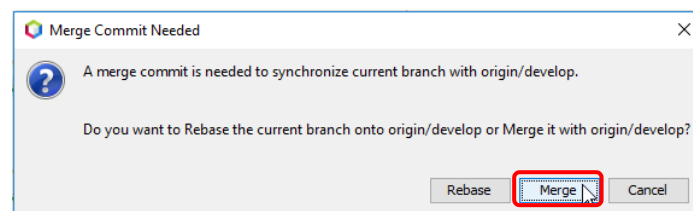
Une belle illustration ici de la sécurité que l'on a lorsque l'on utilise un système de gestion de version puisqu'ici, si l'on avait envoyé nos modifications sur le dépôt distant on aurait écrasé le travail fait par le collaborateur... La solution consiste donc à récupérer (**"Pull"**) les modifications faites sur le dépôt distant et à les fusionner en local (**"Merge"**) avant de pouvoir les envoyer sur le dépôt distant (**"Push"**).

Faites donc à présent un **"Pull"**, on peut alors voir que c'est bien une mise à jour qui est proposée (**"U"**) :



On notera que si l'on avait voulu récupérer la branche **"realisation"** on aurait pu le faire en même temps.

À la suite de ce **"Pull"**, le système de gestion de version vous propose soit de faire un **"Rebase"**, soit de faire un **"Merge"**. Cliquez sur **"Merge"** pour effectuer une fusion avec la branche locale :

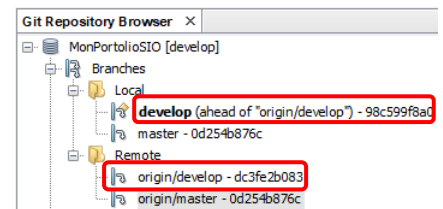


Pourquoi choisir **"Merge"** ? Parce qu'ici on souhaite conserver l'historique complet des changements de branches. En faisant un **"Rebase"**, le résultat sur notre code sera le même, mais l'historique du projet sera réécrit pour donner l'impression que tout a été fait sur la même branche :

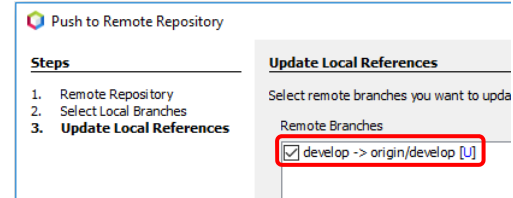
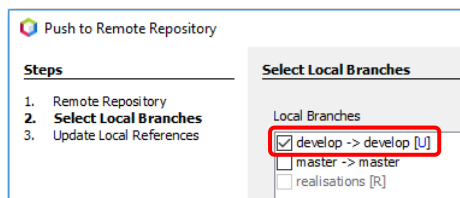
Rebase :

Merge :

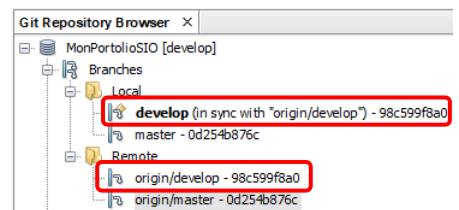
Après ce **"Pull/Merge"**, on peut voir que la version locale est différente de la distante (la version indiquée en local et celle résultante du Merge) :



Faites à présent un **"Push"** de la branche **"develop"**. L'indicateur est à **"U"** pour mise à jour, le **"Push"** devrait cette fois-ci aboutir :



On peut ensuite voir que nos versions locale et distante sont à présent identiques :



5.6 La gestion des conflits

Cette fusion était la plus facilitée car elle pouvait être automatisée sans risque. Les modifications n'affectaient pas les mêmes ressources. Mais qu'en est-il lorsque les modifications concernent les mêmes ressources ?

Retournez sur l'IDE du collaborateur et ajoutez au CV la rubrique "Langues" (il n'y a normalement encore rien dans la "partie droite") :

HTML

```

...
<body>
...
<!-- Partie Droite -->
<div id="langues" class="rubrique clearfix">
  <h2><span class="fa fa-flag"></span> Langue</h2>
  <div class="container">
    <div class="langue">Français</div>
    <div class="niveau">
      <span class="badge badge-dark">A1</span>
      <span class="badge badge-dark">A2</span>
      <span class="badge badge-dark">B1</span>
      <span class="badge badge-dark">B2</span>
      <span class="badge badge-dark">C1</span>
      <span class="badge badge-success">C2</span>
    </div>
  </div>
  <div class="container">
    <div class="langue">Anglais</div>
    <div class="niveau">
      <span class="badge badge-dark">A1</span>
      <span class="badge badge-dark">A2</span>
      <span class="badge badge-dark">B1</span>
      <span class="badge badge-dark">B2</span>
      <span class="badge badge-success">C1</span>
    </div>
  </div>

```

```

        <span class="badge badge-dark">C2</span>
      </div>
    </div>
    <div class="container">
      <div class="langue">Portugais</div>
      <div class="niveau">
        <span class="badge badge-dark">A1</span>
        <span class="badge badge-dark">A2</span>
        <span class="badge badge-dark">B1</span>
        <span class="badge badge-dark">B2</span>
        <span class="badge badge-success">C1</span>
        <span class="badge badge-dark">C2</span>
      </div>
    </div>
    <div class="container">
      <div class="langue">Espagnol</div>
      <div class="niveau">
        <span class="badge badge-dark">A1</span>
        <span class="badge badge-dark">A2</span>
        <span class="badge badge-success">B1</span>
        <span class="badge badge-dark">B2</span>
        <span class="badge badge-dark">C1</span>
        <span class="badge badge-dark">C2</span>
      </div>
    </div>
  </div>
  ...
</body>
</html>

```

Dans le fichier "cv.css", ajoutez ensuite ce contenu :

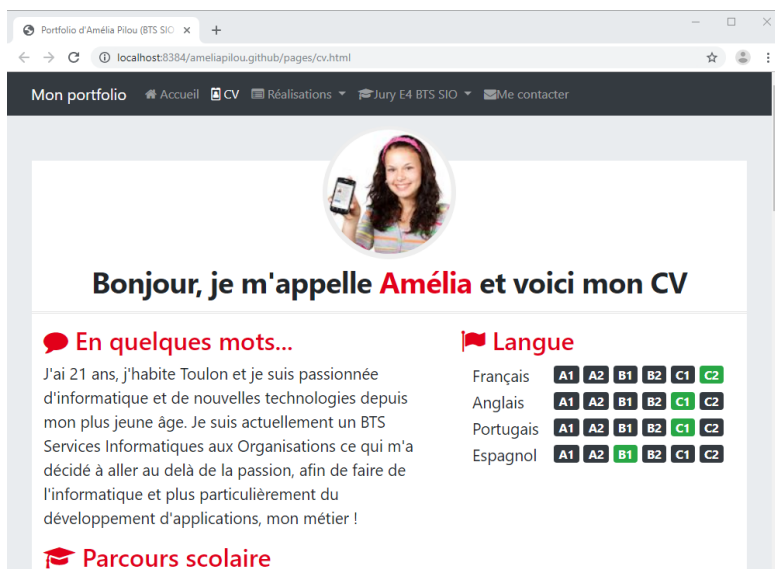
CSS

```

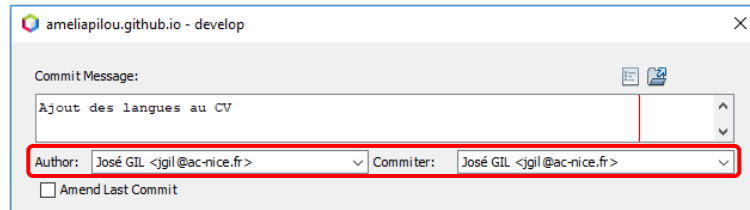
/*****
 *   Langues
 *****/
#langues .container { display: flex; flex-direction: row; justify-content: space-between; }
#langues .langue, #langues .niveau { font-size: 1.3em; }
#langues .langue { padding-top: 3px; }

```

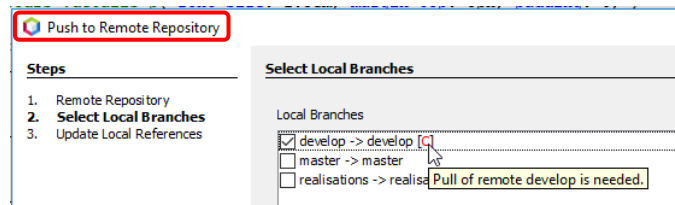
Vous devriez obtenir le visuel suivant :



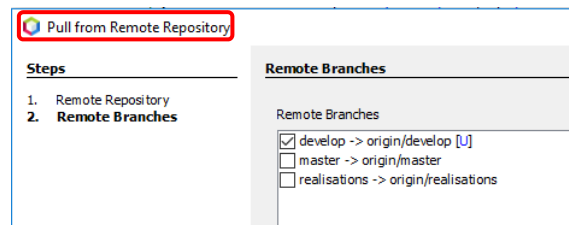
Une fois ces modifications effectuées, faites un nouveau commit :



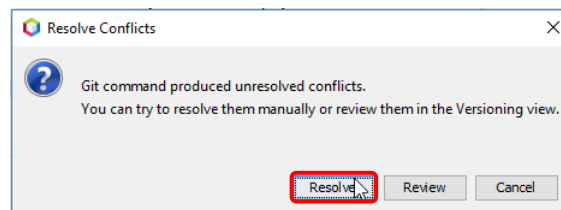
Faites un **"Push"**, vous devriez avoir une alerte **"C"** vous indiquant que ça ne va pas être possible car la version en ligne comporte des modifications que vous n'avez pas en local. Annulez l'opération en cliquant sur **"Cancel"** :



Faites un **"Pull"** pour rapatrier en local les modifications présentes sur le dépôt distant :



Un message apparaît alors vous informant de la présence de conflits. Cliquez sur **"Resolve"** pour les résoudre :



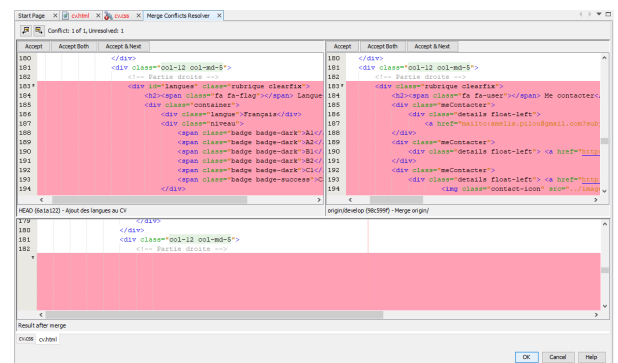
De nouveau une belle illustration de la sécurité que l'on a lorsque l'on utilise un système de gestion de version puisqu'ici, si l'on avait récupéré les modifications sur le dépôt distant on aurait écrasé notre travail... La solution consiste donc à résoudre les conflits en choisissant ce que l'on garde ou pas.

La fenêtre de résolution de conflits se présente ainsi :

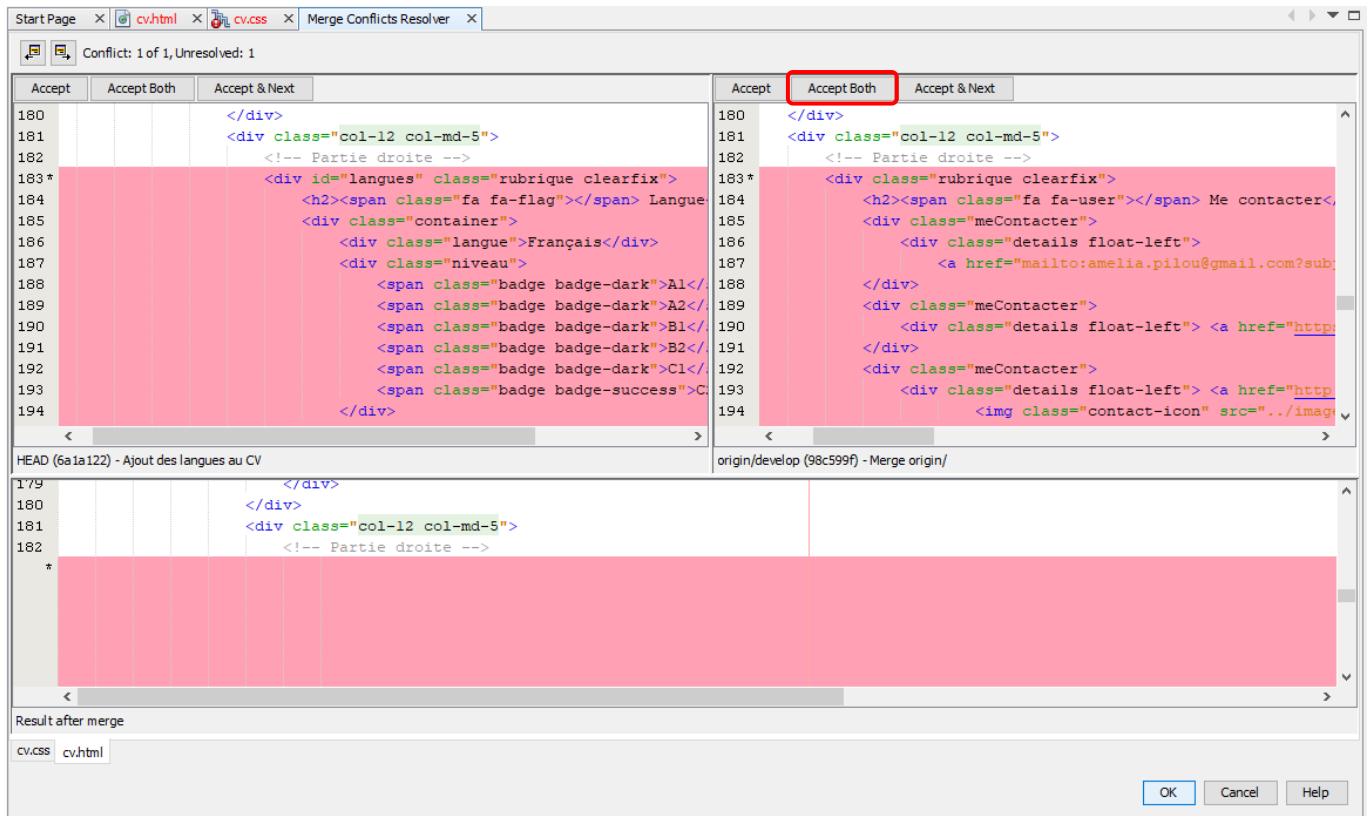
- Dans la partie gauche apparaît le fichier local ;
- Dans la partie droite le fichier distant ;
- En bas, le fichier final

Dans la fenêtre de résolution de conflits :

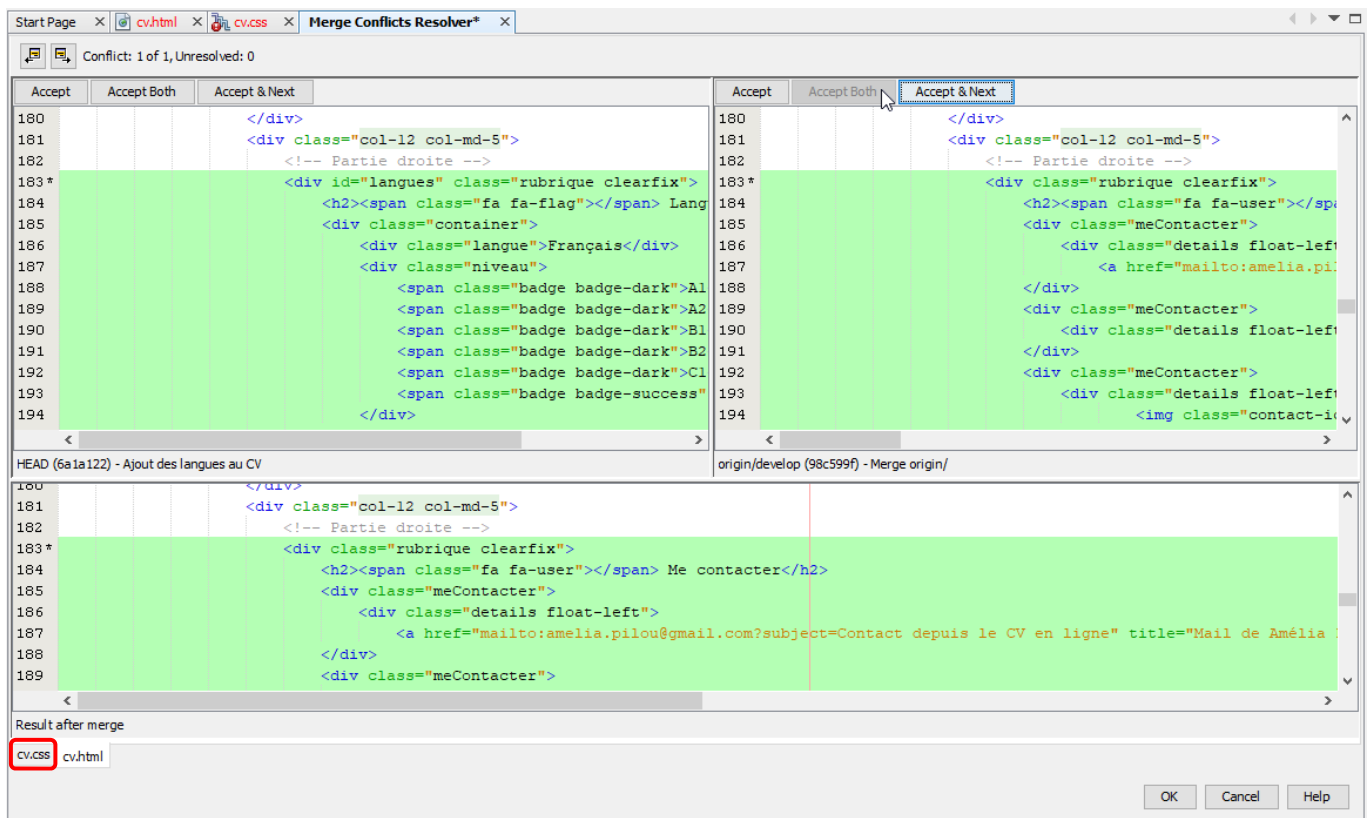
- Pour ne garder que la version locale, il faut cliquer sur le "Accept" de gauche ;
- Pour ne garder que la version distante, il faut cliquer sur le "Accept" de droite ;
- Pour garder les deux cumulée en commençant par la version locale, il faut cliquer sur le "Accept Both" de gauche ;
- Pour garder les deux cumulée en commençant par la version distante, il faut cliquer sur le "Accept Both" de droite.



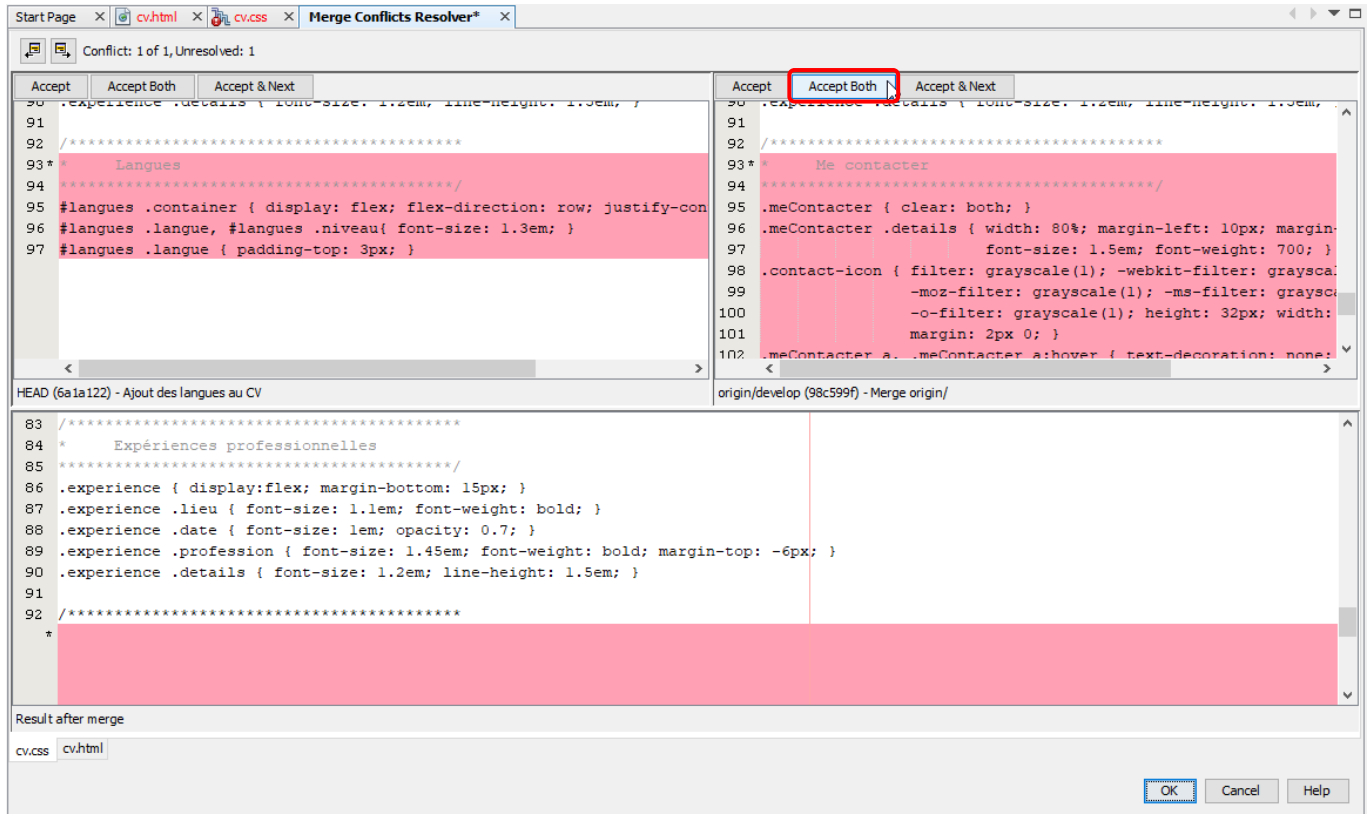
Pour conserver la version distante, et y ajouter la version locale, cliquez sur le bouton **"Accept Both"** de droite :



Les conflits détectés sur cette page passent alors en vert. Cliquez ensuite sur le deuxième fichier (onglet en bas à gauche) pour y résoudre les conflits :

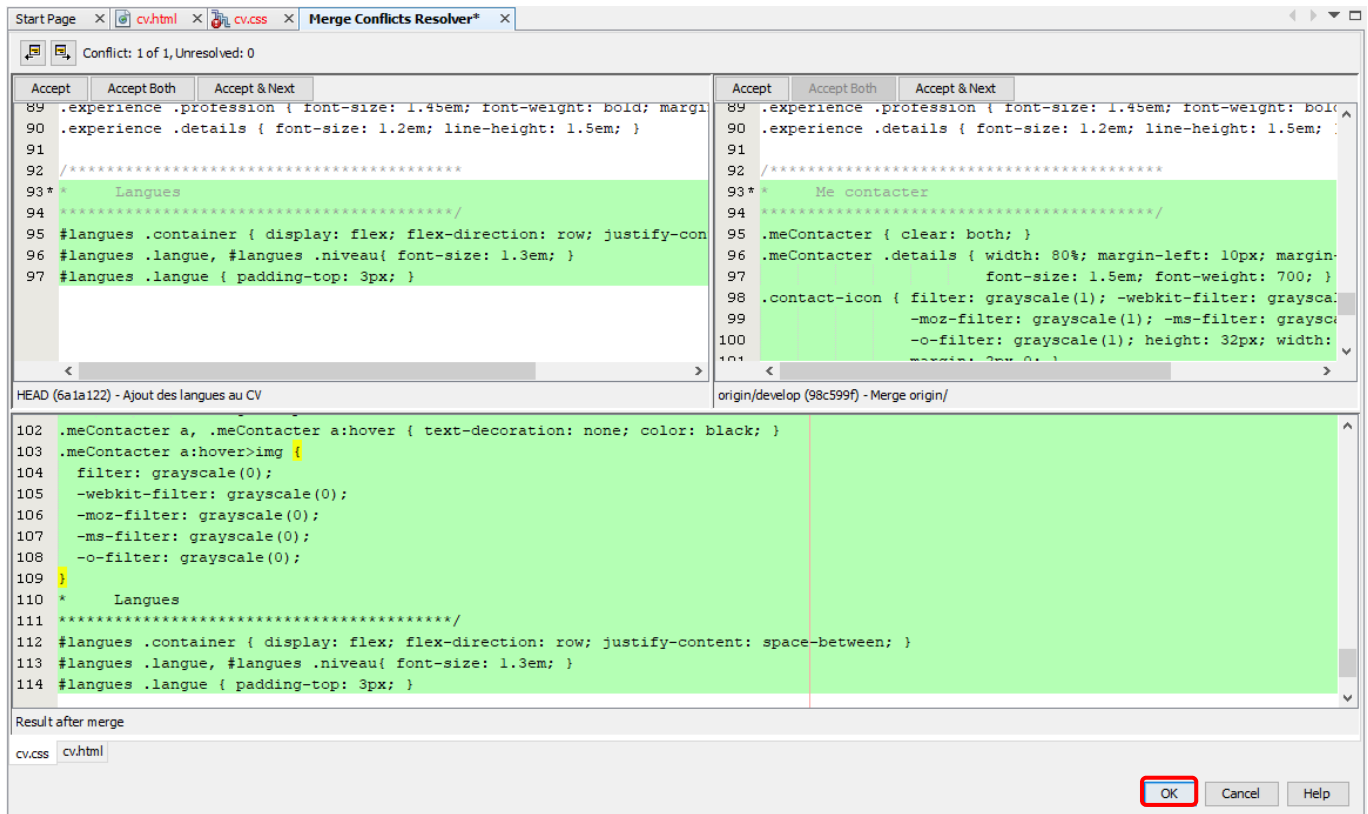


Faites la même chose avec ce second fichier :

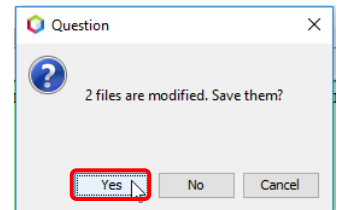


Vous noterez que la solution proposée n'est pas complètement correcte ici puisque la ligne de début de commentaire ici est prise pour la même, au lieu de la voir comme une ligne dupliquée. Ce n'est pas grave, il faudra juste faire un ajustement manuellement après la résolution du conflit.

Cliquez ensuite sur OK pour valider ces résolutions :



Une fenêtre apparaît alors vous proposant de sauvegarder les modifications réalisées pour résoudre le conflit. Cliquez sur "Yes" pour valider la sauvegarde :



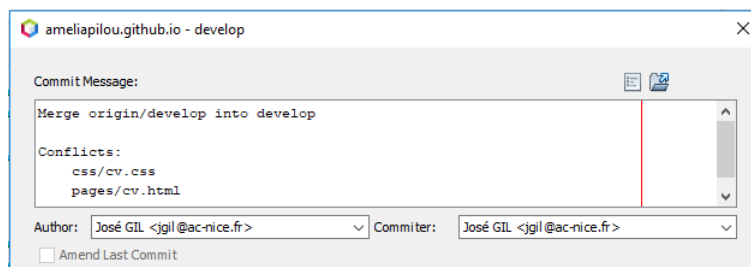
Rendez-vous ensuite dans le fichier cv.css pour ajouter la ligne non dupliquée (avant le commentaire "Langues") :

```

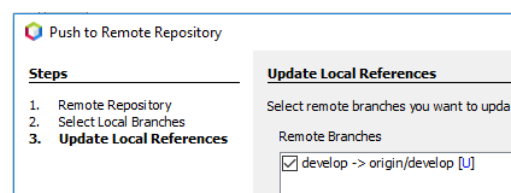
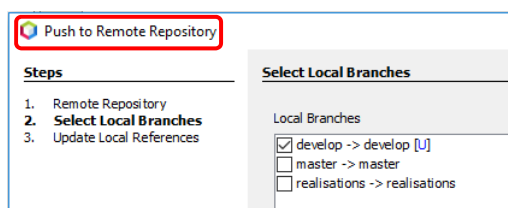
/*****
 * Me contacter
 *****/
.meContacter { clear: both; }
.meContacter .details { width: 80%; margin-left: 10px; margin-top: 3px;
                        font-size: 1.5em; font-weight: 700; }
.contact-icon { filter: grayscale(1); -webkit-filter: grayscale(1);
                -moz-filter: grayscale(1); -ms-filter: grayscale(1);
                -o-filter: grayscale(1); height: 32px; width: 32px;
                margin: 2px 0; }
.meContacter a, .meContacter a:hover { text-decoration: none; color: black; }
.meContacter a:hover>img {
    filter: grayscale(0);
    -webkit-filter: grayscale(0);
    -moz-filter: grayscale(0);
    -ms-filter: grayscale(0);
    -o-filter: grayscale(0);
}
* Langues
*****/
#langues .container { display: flex; flex-direction: row; justify-content: space-between; }
#langues .langue, #langues .niveau { font-size: 1.3em; }
#langues .langue { padding-top: 3px; }

```

Enregistrez vos modifications et faites un "Commit". Le message de commit est alors proposé avec des indications sur la résolution des conflits :



Le code de la branche étant de nouveau stabilisé, nous allons pouvoir faire notre "Push" sans problème :



Retournez sur l'IDE principal (celui d'Amélia dans notre cas) et ajoutez au CV la rubrique "Compétences" :

HTML

```
...
<body>
...
  <!-- Partie Droite -->
  ...
  <div id="competences" class="rubrique clearfix">
    <h2><span class="fa fa-wrench"></span> Compétences</h2>
    <div class="container">
      <div class="libelle">Intégration web</div>
      <div class="classement classement5"></div>
    </div>
    <div class="container">
      <div class="libelle">Dév. web</div>
      <div class="classement classement4"></div>
    </div>
    <div class="container">
      <div class="libelle">Administration</div>
      <div class="classement classement3"></div>
    </div>
  </div>

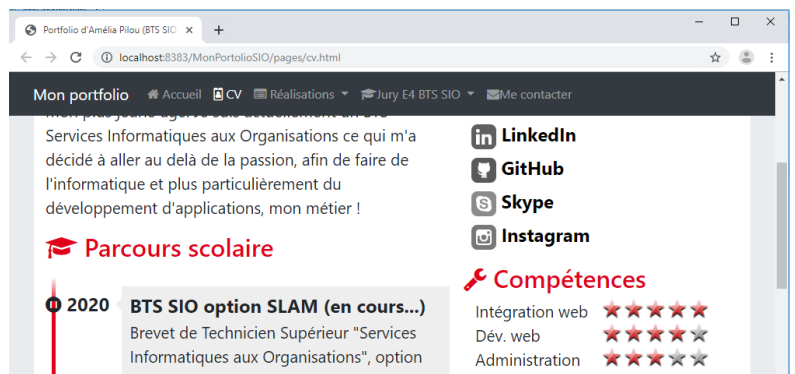
  ...
</body>
</html>
```

Dans le fichier "cv.css", ajoutez ensuite ce contenu :

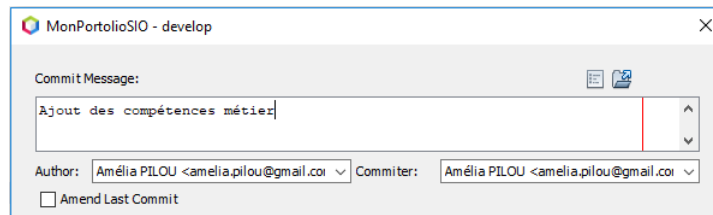
CSS

```
/* *****
*   Compétences
* ***** */
#competences .container {
  display: flex; flex-direction: row; justify-content: space-between; }
#competences .container .libelle { font-size: 1.3em; }
#competences .container .classement {
  background: scroll transparent url(images/Red-Rating-Stars.png) no-repeat;
  width: 160px; height: 27px; }
#competences .container .classement0 { background-position: 0 -142px; }
#competences .container .classement1 { background-position: 0 -112px; }
#competences .container .classement2 { background-position: 0 -84px; }
#competences .container .classement3 { background-position: 0 -56px; }
#competences .container .classement4 { background-position: 0 -28px; }
#competences .container .classement5 { background-position: 0 0px; }
```

Vous devriez obtenir le visuel suivant :

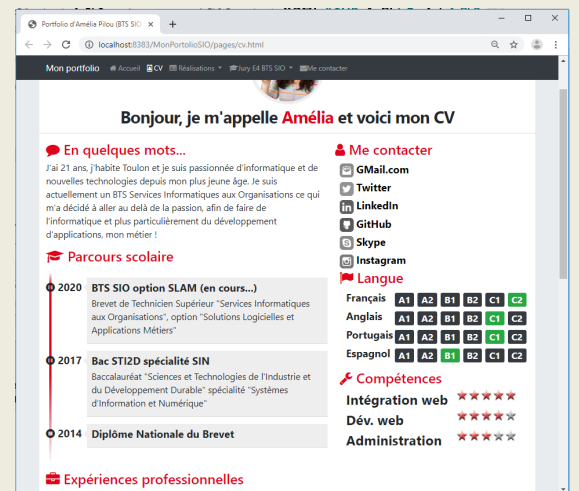


Une fois ces modifications effectuées, faites un nouveau commit :



TRAVAIL À FAIRE

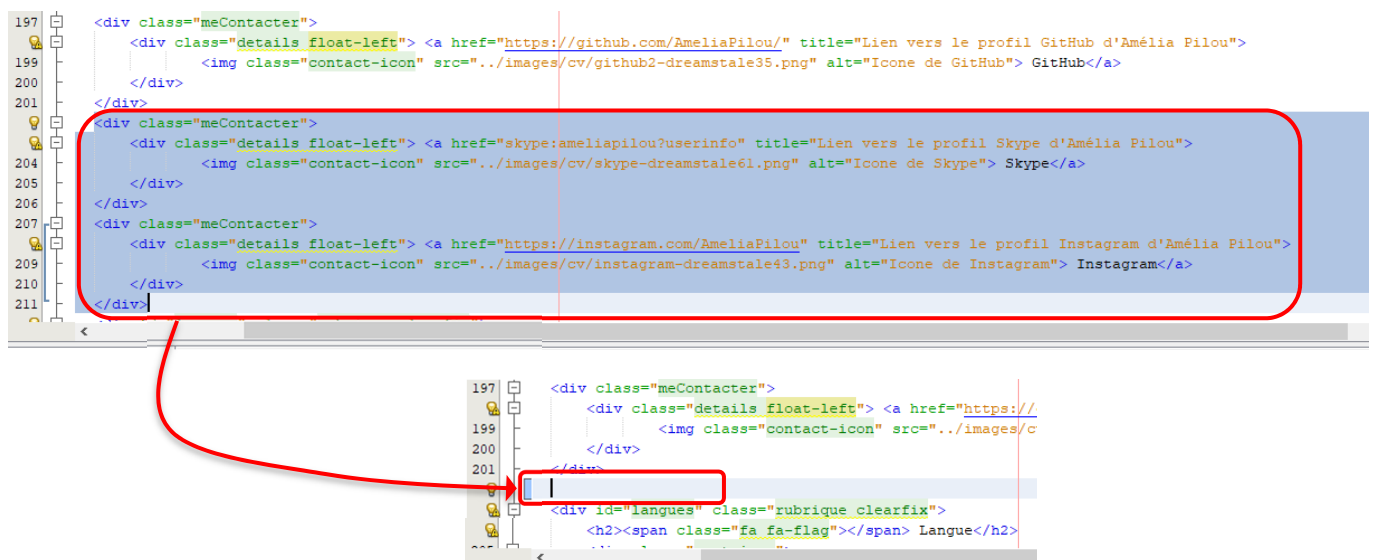
En vous basant sur les opérations réalisées depuis l'IDE du collaborateur, faites-en sorte d'intégrer les modifications locales à la suite de celles présentes sur le dépôt distant afin d'arriver à ce résultat :



5.7 Réparer du code non commité

Comme on l'a vu dans le cours, l'utilisation d'un outil de gestion de version permet de travailler sereinement, sans avoir peur de faire des modifications.

Dans le CV, supprimez les profils Skype et Instagram, enregistrez le fichier (pas de commit) et redémarrez NetBeans :



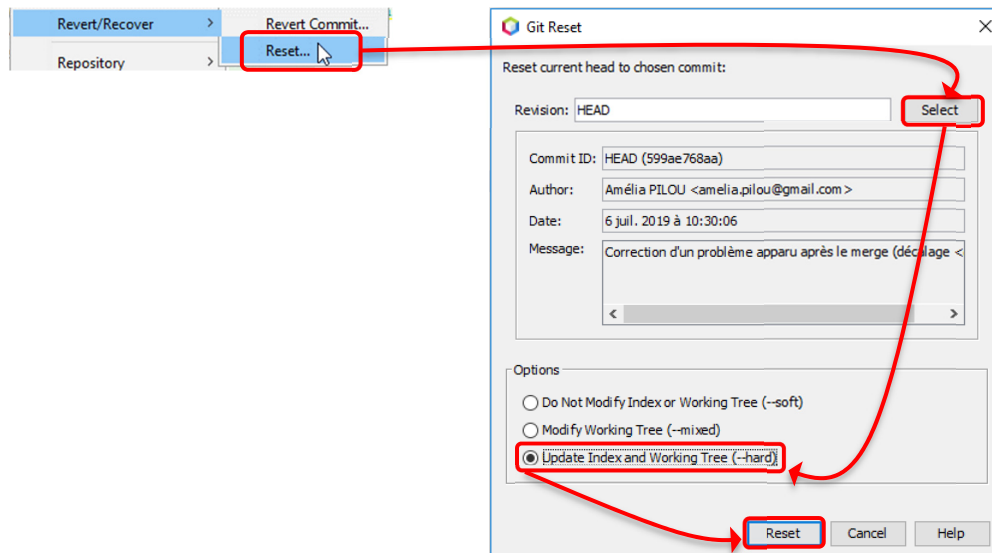
Dans ce genre de cas, on se dit toujours que l'on peut revenir en arrière à coup de "Ctrl+Z"... Oui, c'est vrai, mais à condition de n'avoir pas fermé l'IDE ! Une fois NetBeans relancé, on peut voir qu'il n'y a plus rien à annuler !

Plusieurs possibilités s'offrent à vous dans NetBeans :

- Soit vous rétablissez juste certaines portions de codes manquantes en cliquant sur la zone signalée par Git et en cliquant sur la flèche "Replace with original text" :



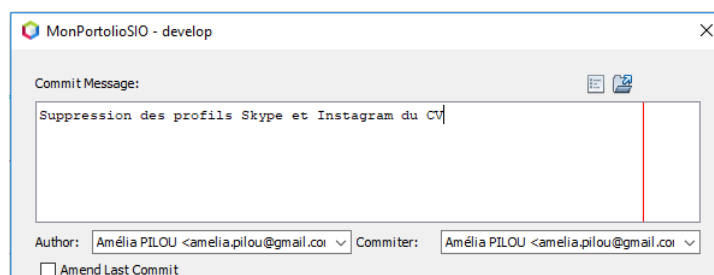
- Soit vous rétablissez carrément la dernière version du projet (ou un autre) en cliquant sur "**Git ▶ Revert/Recover ▶ Reset**". Sur la fenêtre de reset, si vous voulez rétablir une autre version cliquez sur "**Select**" et choisissez la version à rétablir, sinon choisissez directement l'option "**Update Index and Working Tree**" (ce qui permet de forcer l'actualisation du fichier) et cliquez sur "**Reset**" :



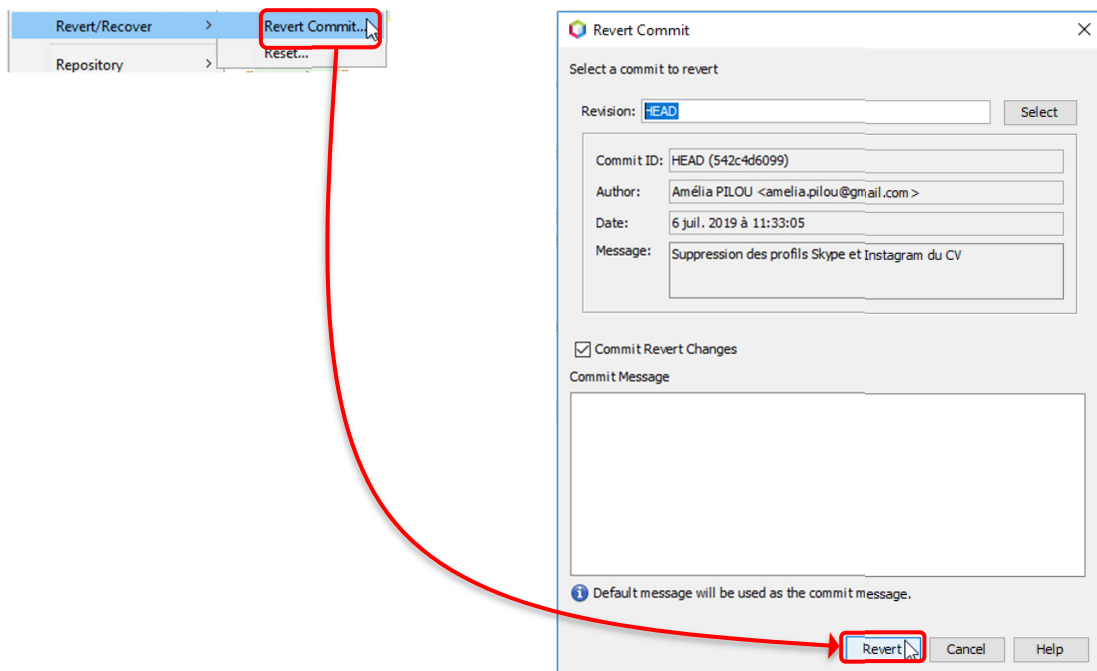
Le fichier revient tel qu'il était au précédent commit !

5.8 Réparer du code commité

Pour cette démonstration, supprimez de nouveau les profils Skype et Instagram du CV, mais cette fois-ci procédez à un commit :



Pour annuler ce commit, on va procéder à un **"Revert"** de ce commit, c'est-à-dire que l'on va en annuler les effets. Cliquez sur le menu **"Git ► Revert/Recover ► Revert Commit"**. Sur la fenêtre de revert, le dernier commit vous est proposé, cliquez sur **"Revert"** pour annuler les effets de ce commit :



Dans cet exemple on a pris le dernier commit, mais un revert peut s'appliquer sur n'importe quel commit. Rappelons-nous qu'un commit n'est ni plus ni moins qu'un **"patch"** avec une liste d'actions effectuées qu'il est donc possible de "défaire".

Vous pouvez d'ailleurs voir le patch qui a été utilisé pour faire ce revert en cliquant sur le menu **"Team ► Patches ► Export commit"** :

```

Start Page x indexhtml x accueil.css x cv.css x cv.html x appliandroid.html x revert.patch x
5  Revert "Suppression des profils Skype et Instagram du CV"
6
7  This reverts commit 542c4d60994848ce26f9ea4bc86bb80305c78305.
8
9  diff --git a/pages/cv.html b/pages/cv.html
10 index f744220..7e8b467 100644
11 --- a/pages/cv.html
12 +++ b/pages/cv.html
13 @@ -199,7 +199,16 @@
14
15         </div>
16     </div>
17 -
18 +     <div class="meContacter">
19 +         <div class="details float-left"> <a href="skype:amelia.pilou?userinfo" title="Lien vers le profil Skype d'
20 +          Skype</
21 +         </div>
22 +     </div>
23 +     <div class="meContacter">
24 +         <div class="details float-left"> <a href="https://instagram.com/AmeliaPilou" title="Lien vers le profil I
25 +         
26 +         </div>
27 +     </div>
28 +
29     <div id="langues" class="rubrique clearfix">
30         <h2><span class="fa fa-flag"></span> Langue</h2>
31         <div class="container">

```

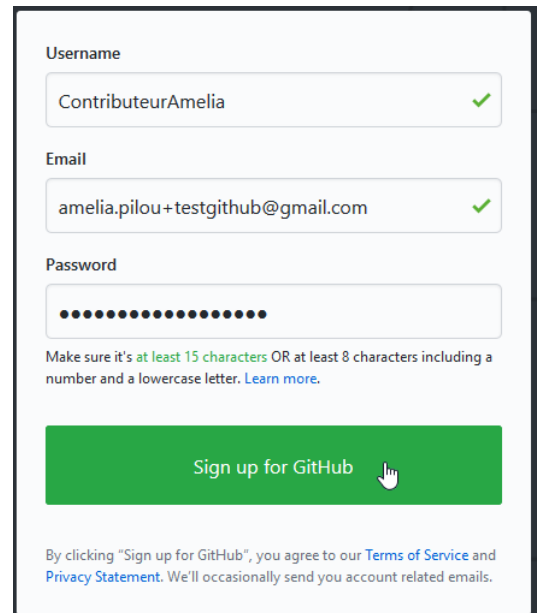
TRAVAIL À FAIRE

Terminez le CV en y ajoutant une rubrique "Passions" et fusionnez la branche "develop" avec la branche "master" pour pouvoir ensuite mettre automatiquement en ligne le CV terminé sur les pages GitHub.

5.9 Collaborer sans être collaborateurs

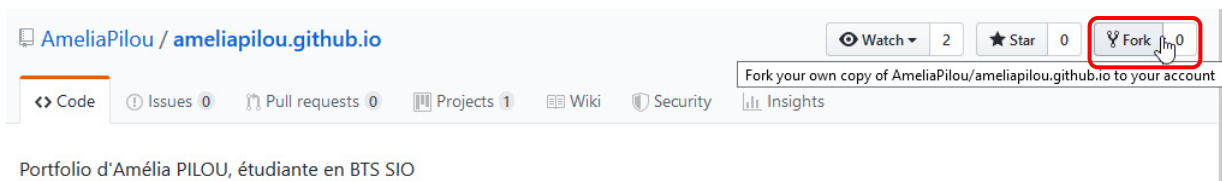
Nous avons vu comment collaborer dans GitHub en indiquant dans les paramètres des dépôts les collaborateurs autorisés. Dans une petite équipe, cela peut être pratique, mais dans le cas où l'on est nombreux, cela peut vite être compliqué à gérer. De plus cela n'offre pas la possibilité de collaborer ponctuellement à des projets dans lesquels on ne fait pas forcément partie des équipes de développement : bénévolat, projets open source, projets communautaires...

Pour cet exemple, créez un nouveau compte GitHub (en utilisant une des astuces Gmail vues précédemment) :



The screenshot shows the GitHub sign-up page. It has fields for Username (ContributeurAmelia), Email (amelia.pilou+testgithub@gmail.com), and Password (masked with dots). There is a green 'Sign up for GitHub' button. Below the button, it says: 'By clicking "Sign up for GitHub", you agree to our [Terms of Service](#) and [Privacy Statement](#). We'll occasionally send you account related emails.'

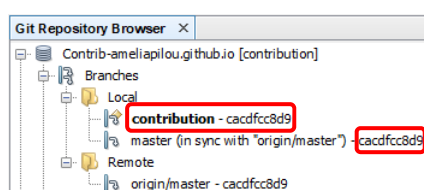
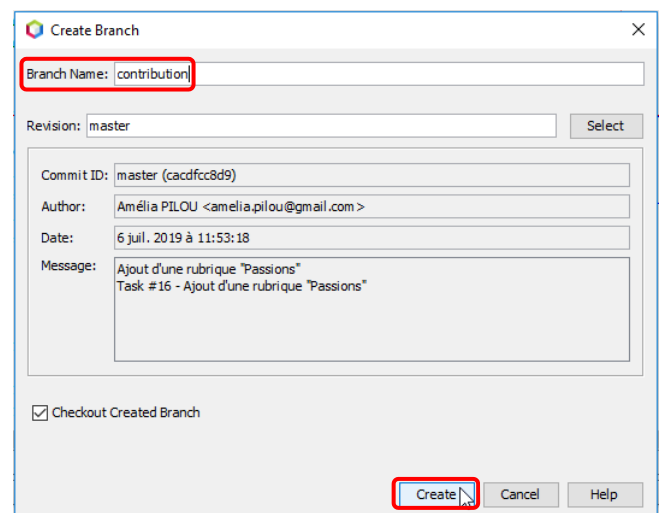
Avec ce nouveau compte, rendez-vous ensuite sur le dépôt principal, puis cliquez sur **"Fork"** :



Dans le dépôt du contributeur, on voit alors apparaître ce nouveau dépôt comme un "fork" du dépôt principal :



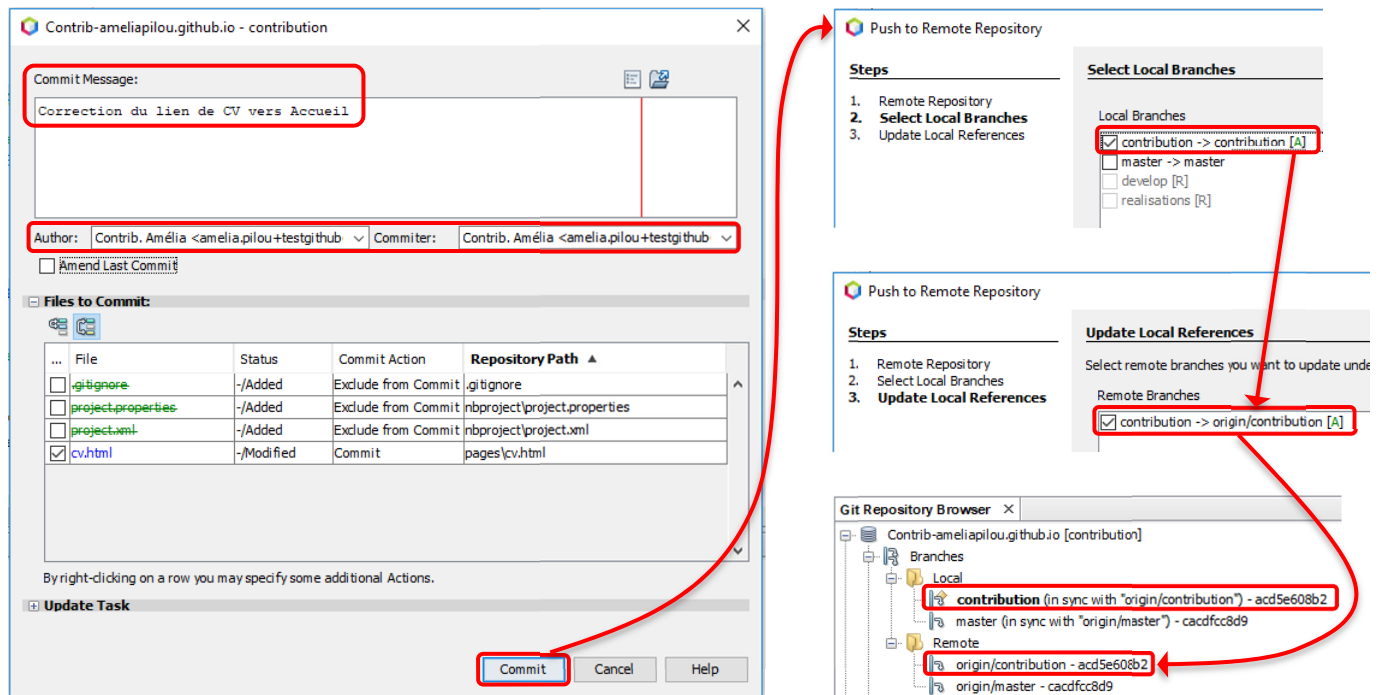
Par convention, lorsque l'on veut contribuer à un projet, on n'intervient pas sur la branche principale, comme sur son propre projet d'ailleurs (voir précédemment) ... Vous allez donc créer une branche **"contribution"** basée sur la branche **"master"** :



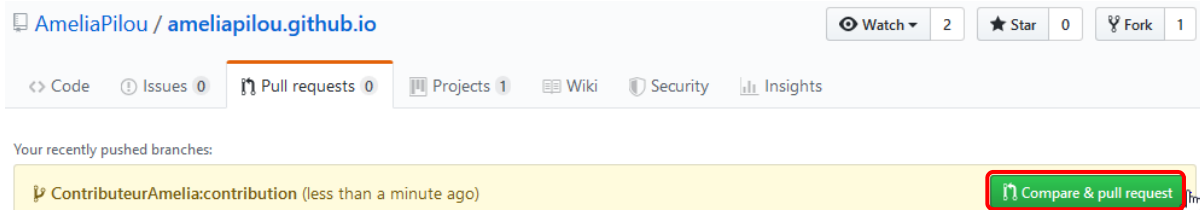
Faites une modification dans le projet. Par exemple en corrigeant une erreur (il y en a normalement plusieurs dans le projet) comme sur la page de CV où le lien vers la page d'accueil est erroné :

```
<div id="navbar" class="navbar-collapse collapse">
  <ul class="nav navbar-nav">
    <li class="nav-item">
      <a href="../../index.html" class="nav-link"><span class="fa fa-home"></span> Accueil</a>
    </li>
  </ul>
</div>
```

Faites ensuite un **"Commit"** (veillez à bien renseigner les informations du contributeur comme **"Author"** et **"Committer"**) et un **"Push"** sur le dépôt distant (celui du contributeur) :



Retournez sur le dépôt principal. GitHub voit que vous avez fait un "Push" et vous propose directement de comparer votre branche avec le projet en cours pour ensuite faire une "Pull Request". Cliquez sur le bouton **"Compare & pull request"** :



Une "pull request" se traduit littéralement par une "demande de récupération". Dans le vocabulaire GitHub, on parle souvent de "PR". Il s'agit de demander simplement à autre développeur de faire un pull d'une branche de votre dépôt vers le sien. Pour ce faire il suffit d'avoir les informations suivantes :

- Dépôt source / Branche source
- Dépôt cible / Branche cible.

Dans cette fenêtre de "Pull request", on peut voir que votre branche est prête pour une fusion avec la branche principale (pas de conflit détecté). On peut également voir que le titre du message a été automatiquement repris du dernier commit. Par convention, lorsque l'on fait une "Pull request" il y a quelques règles à respecter :

- **Rédiger un message clair et courtois** : Dans toute interaction la politesse doit toujours être de rigueur. Vous pouvez également profiter de ce message pour remercier le développeur ou la développeuse pour le code partagé sur GitHub.
- **Être "autorisé" à travailler sur le projet** : Dans le cadre d'un projet open source c'est souvent implicite, mais dans les autres cas il peut être parfois utile d'échanger avec le mainteneur du projet via les "Issues" pour savoir si de l'aide est demandée ou appréciée...
- **Faire des "petites" pull requests** : Mieux vaut faire 5 pull requests de 3 modifications, que 1 pull request de 15 modifications.

Vous êtes ensuite redirigé vers l'onglet "Pull requests" du projet où vous pouvez voir votre demande avec le statut "Open" :

Du côté du mainteneur et des membres du projet, un mail est reçu avec l'information de la pull request, et des liens directs vers leur gestion. Cliquez sur le lien pour vous rendre directement sur la "pull request" (avec le compte GitHub principal ou celui du collaborateur, pas celui du contributeur) :

En général avant d'accepter des modifications, on se rend sur l'onglet "Files changed" pour voir quelles sont les modifications apportées dans cette pull request :

Si les modifications conviennent et que l'on souhaite les accepter, il suffit de revenir à l'onglet "Conversation" et cliquer sur le bouton "Merge pull request" pour fusionner la branche proposée avec la nôtre :

The screenshot shows a GitHub pull request interface. At the top, the repository is 'AmeliaPilou / ameliapilou.github.io'. The pull request title is 'Correction du lien de CV vers Accueil #18'. The status bar indicates 'Conversation 0', 'Commits 1', 'Checks 0', and 'Files changed 1'. A comment from 'ContributeurAmelia' is visible, stating: 'Bonjour, J'ai constaté sur le CV que le lien pointant vers la page d'accueil était erroné. Vous trouverez ma proposition de correction. Bien cordialement, Contri Buteur'. Below the comment, a commit titled 'Correction du lien de CV vers Accueil' is shown. A green box highlights the status 'This branch has no conflicts with the base branch' and a red box highlights the 'Merge pull request' button. The right sidebar shows various settings like 'Reviewers', 'Assignees', 'Labels', 'Projects', 'Milestone', and 'Notifications'.

Il faut ensuite cliquer sur le bouton "Confirm merge" pour que la fusion soit effective :

The screenshot shows a 'Confirm merge' dialog box. It contains the text 'Merge pull request #18 from ContributeurAmelia/contribution' and 'Correction du lien de CV vers Accueil'. At the bottom, there are two buttons: 'Confirm merge' (highlighted with a red box) and 'Cancel'.

Le statut de la pull request passe alors à "Merged". Cette étape peut également être l'occasion pour le mainteneur de remercier le contributeur :

AmeliaPilou / ameliapilou.github.io

Unwatch 2 Star 0 Fork 1

Code Issues 0 Pull requests 1 Projects 1 Wiki Security Insights Settings

Correction du lien de CV vers Accueil #18

Merged AmeliaPilou merged 1 commit into AmeliaPilou:master from ContributeurAmelia:contribution 20 seconds ago

Conversation 0 Commits 1 Checks 0 Files changed 1 +1 -1

ContributeurAmelia commented 7 minutes ago

Bonjour,

J'ai constaté sur le CV que le lien pointant vers la page d'accueil était erroné. Vous trouverez ma proposition de correction.

Bien cordialement,
Contri Buteur

Correction du lien de CV vers Accueil acd5e60

AmeliaPilou merged commit bc7d991 into AmeliaPilou:master 20 seconds ago

Write Preview

Merci de votre contribution 🙌👍

Attach files by dragging & dropping, selecting or pasting them.

Comment

Reviewers: No reviews

Assignees: No one—assign yourself

Labels: None yet

Projects: None yet

Milestone: No milestone

Notifications: You're receiving notifications because you're watching this repository. Unsubscribe

2 participants

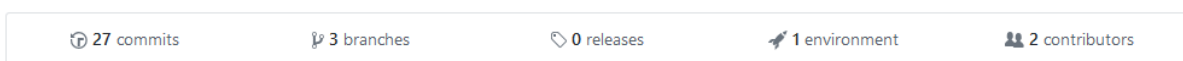
Beaucoup d'entreprises privilégient, y compris en interne, les pull requests car ils permettent un meilleur suivi des modifications et favorisent les "Revue de code".

6. Utilisation avancée de GitHub

GitHub dispose d'une interface facile à prendre en main et qui facilite la communication en donnant accès rapidement à des fonctionnalités et des informations intéressantes.

6.1 Informations sur le cycle de vie d'un projet

Sur la page d'accueil d'un dépôt sont regroupées beaucoup d'informations : Nombre de commits, nombre de branches, nombre de contributeurs...

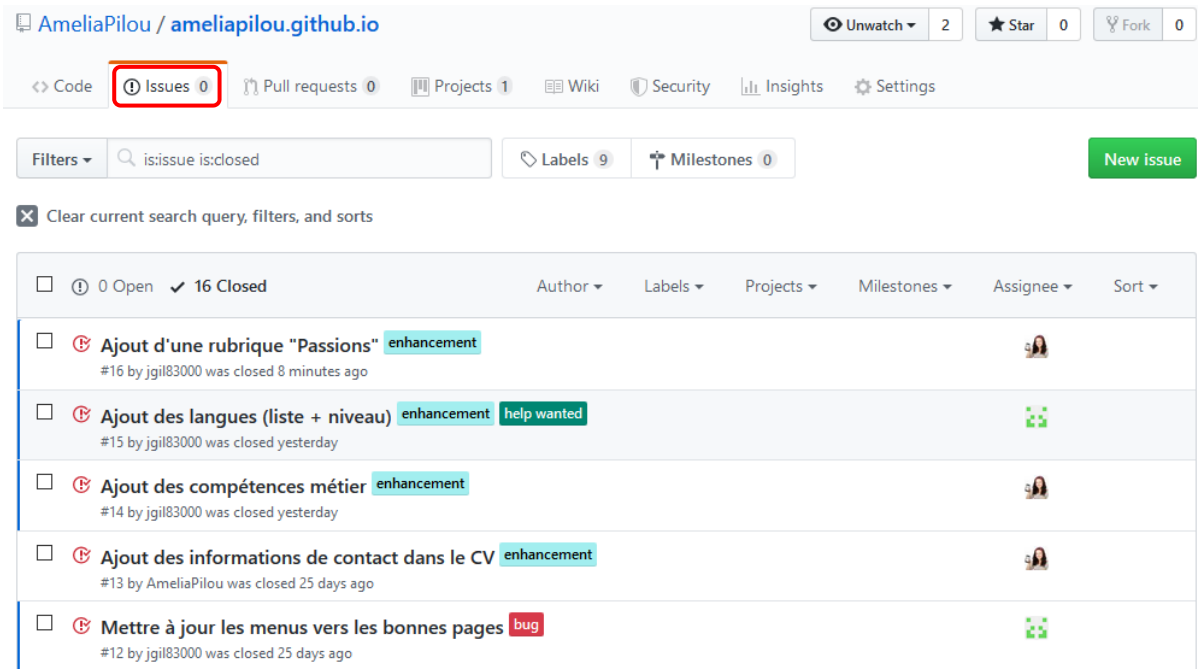


Et en cliquant sur la barre de couleur, on peut même avoir la proportion générale de langages utilisés :



6.2 Communication et gestion des incidents

GitHub, comme toute bonne forge en ligne, offre la possibilité de reporter des problèmes et poser des questions aux collaborateurs du projet. Il faut pour cela se positionner sur la rubrique **"Issues"**. Pour l'instant dans notre projet on s'est servi des "issues" pour le suivi du projet, mais l'objectif principal de cet outil est bien de faire de la gestion d'incidents (demande de nouvelles fonctionnalités, report de bugs...) :



AmeliaPilou / ameliapilou.github.io

Unwatch 2 Star 0 Fork 0

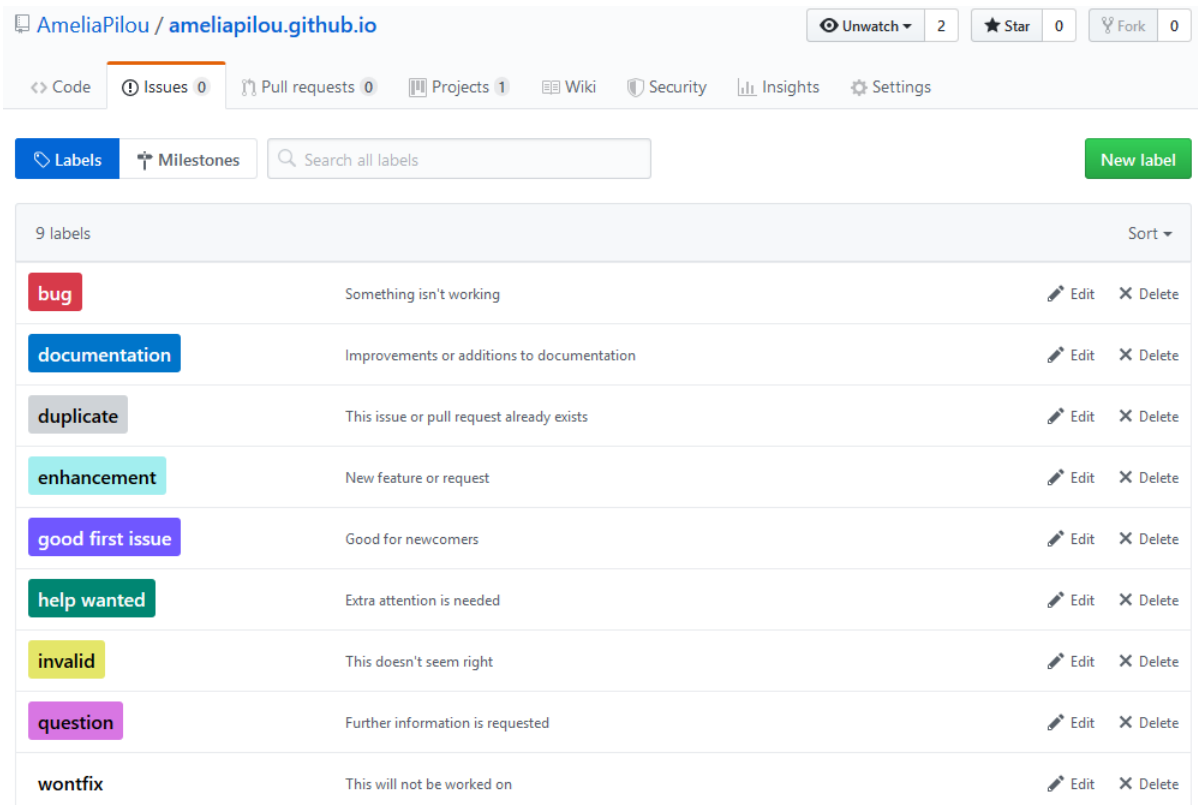
Code Issues 0 Pull requests 0 Projects 1 Wiki Security Insights Settings

Filters is:issue is:closed Labels 9 Milestones 0 New issue

Clear current search query, filters, and sorts

☐	0 Open ✓ 16 Closed	Author	Labels	Projects	Milestones	Assignee	Sort
☐	Ajout d'une rubrique "Passions" enhancement #16 by jgil83000 was closed 8 minutes ago						
☐	Ajout des langues (liste + niveau) enhancement help wanted #15 by jgil83000 was closed yesterday						
☐	Ajout des compétences métier enhancement #14 by jgil83000 was closed yesterday						
☐	Ajout des informations de contact dans le CV enhancement #13 by AmeliaPilou was closed 25 days ago						
☐	Mettre à jour les menus vers les bonnes pages bug #12 by jgil83000 was closed 25 days ago						

Comme un fil de forum, les messages postés permettent aux membres du projet de communiquer et de garder un historique. Les messages postés sont catégorisables, par défaut vous disposez de plusieurs catégories identifiées par des couleurs (les plus courants dans les projets) mais il est toujours possible de les personnaliser (par exemple pour ceux qui veulent des labels plus précis, ou francisés) :



AmeliaPilou / ameliapilou.github.io

Unwatch 2 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 1 Wiki Security Insights Settings

Labels Milestones Search all labels New label

9 labels Sort

bug	Something isn't working	Edit Delete
documentation	Improvements or additions to documentation	Edit Delete
duplicate	This issue or pull request already exists	Edit Delete
enhancement	New feature or request	Edit Delete
good first issue	Good for newcomers	Edit Delete
help wanted	Extra attention is needed	Edit Delete
invalid	This doesn't seem right	Edit Delete
question	Further information is requested	Edit Delete
wontfix	This will not be worked on	Edit Delete

Le message une fois posté peut être réédité, peut recevoir des réponses et peut être clos (lorsque la question ou le problème est résolu) :

L'avantage est que chaque fois qu'un message est posté, les membres du projet sont automatiquement avertis par mail ce qui leur permet d'avoir une plus grande réactivité :

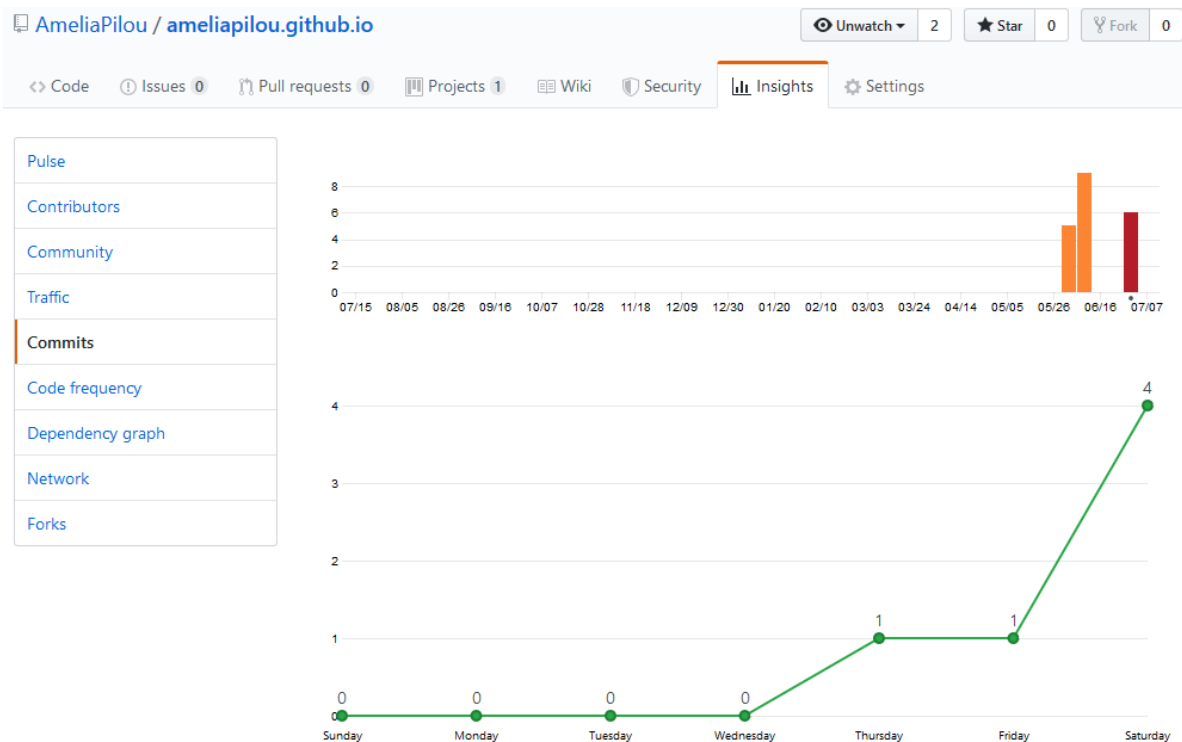
6.3 Alimenter le wiki du projet

GitHub offre également la possibilité d'alimenter un wiki. Pour cela, il faut se rendre dans la rubrique **"Wiki"** et créer de nouvelles pages :

The screenshot shows the GitHub interface for the repository 'AmeliaPilou / ameliapilou.github.io'. The 'Wiki' tab is selected, displaying a page titled 'Rendre un onglet "actif" avec Bootstrap'. The page content explains that to make a navbar item active, the 'active' class must be added to the list item. It shows the code change from `<li>` to `<li class="active">`. On the right, a sidebar shows the 'Pages' section with a search bar and a list of pages, including 'Home' and the current page. There are also buttons for 'Edit' and 'New Page'.

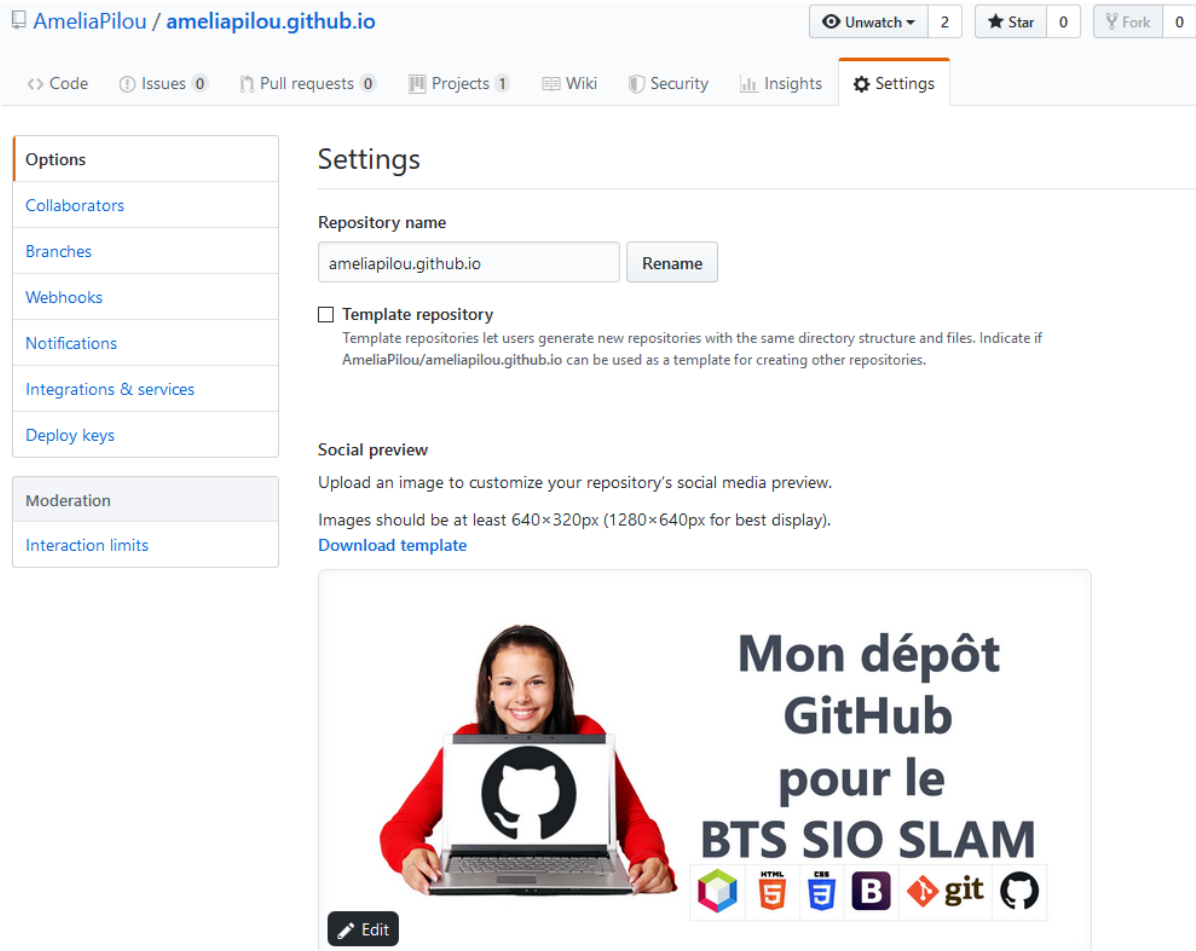
6.4 Indicateurs de suivi de projet

GitHub met à disposition des indicateurs de suivi de projet dans la rubrique **"Insights"**. Ils permettent de suivre l'activité du projet (le nombre de développeurs, le nombre de validations, la fréquence des évolutions...) :



6.5 Administration du projet

La dernière rubrique permet d'administrer le projet. C'est par exemple dans cette partie que l'on va pouvoir renommer un projet, ajouter des collaborateurs, utiliser des WebHooks prédéfinis (Classes prédéfinies permettant d'utiliser certaines API) pour communiquer avec d'autres plateformes (messageries instantanées, email, autres gestionnaires de projets...) :



Cette initiation est maintenant terminée, il ne vous reste plus qu'à utiliser régulièrement les outils que vous avez découverts pour vous perfectionner et pour acquérir une pratique plus proche de ce qui se fait en entreprise.

La logique de gestion de version étant maintenant assimilée, n'hésitez pas à rechercher des solutions vous permettant de les utiliser directement dans vos autres IDE :

- Par exemple pour les utilisateurs d'Eclipse, il existe le plug-in « eGit » (www.eclipse.org/egit) qui permet de versionner les projets développés sous Eclipse directement depuis l'IDE.
- Et dans les dernières versions de Microsoft Visual Studio tous les outils pour travailler avec Git et GitHub sont intégrés (d'autant plus que GitHub appartient à Microsoft depuis juin 2018).

N'hésitez pas non plus à tester d'autres forges logicielles telles que Microsoft Team Foundation Server (remplacé depuis mars 2019 par "Azure DevOps Server"), BitBucket, ou GitLab.com (qui présente l'avantage de s'appuyer sur GitLab qui est une plateforme open source que vous pouvez utiliser sur vos propres serveurs).