# # Comment se mettre à GitHub Pages et Jekyll ?

## ## Résumé

Si comme moi, vous avez envie de monter votre blog sans vous compliquer la vie, ce mini-tutoriel est pour vous. Il vous présentera en quelques actions comment mettre en place un blog en s’appuyant sur du déjà tout prêt. A savoir :

* GitHub pour l’hébergement des sources
* GitHub Pages pour l’accessibilité du site Web
* Jekyll pour la mise en place d’un blog structuré en quelques clics

A vos claviers !

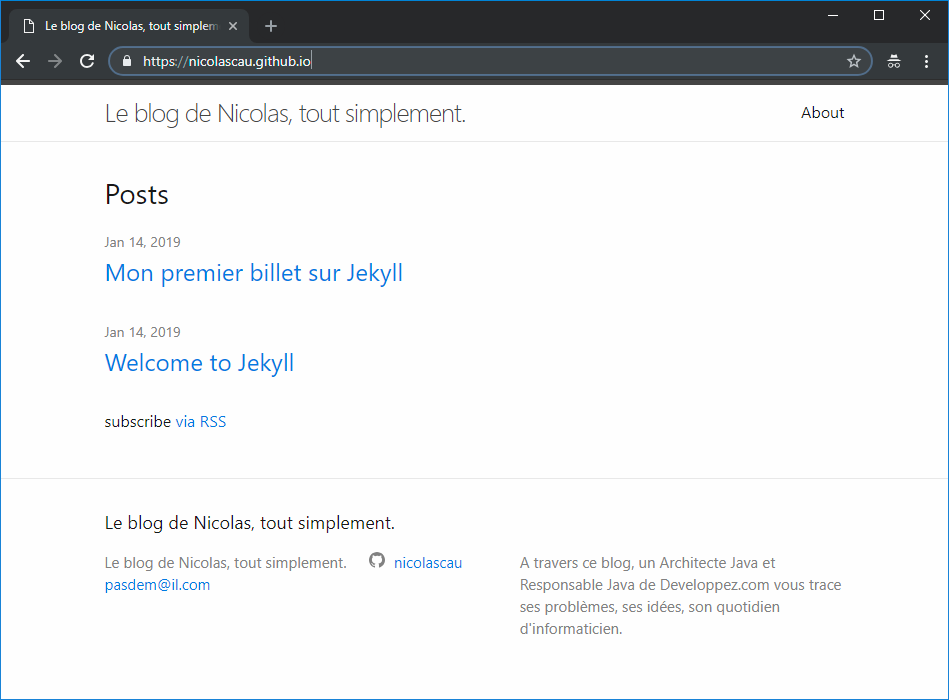
## ## Introduction

Depuis quelques semaines, j’ai eu envie de me mettre en place un blog. Non pas que j’ai envie de vous raconter ma vie. Néanmoins, je me dis que c’est une façon de capitaliser mes connaissances d’informaticien. Je serais donc mon premier lecteur (c’est déjà ça 😉) !

Mon besoin est donc simple :

* Disposer d’un site Web sans me soucier de la façon de l’héberger
* Disposer d’un blog structuré et très facile à configurer/modifier/adapter

Et si vous me suivez, vous arriverez à ceci :



Superbe, n’est-ce pas ?

## ## Prérequis

Pour commencer tout cela, je me suis déjà créé un compte GitHub. Sans compte GitHub, vous n’irez pas plus loin. Pour créer votre compte, dirigez-vous sur [Github](https://github.com/).

Ensuite, j’ai aussi besoin que vous ayez quelques connaissances :

* Sur Git
* Sur GitHub (mais bon si vous y avez déjà un compte, c’est surement le cas)
* Sur le format Markdown (mais vraiment un tout petit peu)

Je base aussi mes explications sur un environnement Windows.

## ## 1ère étape : GitHub Pages pour héberger

[GitHub Pages]( https://pages.github.com/) est un service d’hébergement de site statique. Il présente l’avantage d’être directement connecté à votre repository GitHub. Grâce à ce système, vous pouvez créer des sites Web.

Deux points sont à conserver en tête lors de sa mise en place :

* GitHub Pages est limité (pas de langages côté serveur, des limites de taille de repo., des limites de bande passante, etc.) N’espérez donc pas héberger votre site en production sur ce repository.
* Un site GitHub Pages est visible en public sur le net. N’allez donc pas y intégrer des données sensibles.

Vous trouverez plus d’information sur les limitations et le fonctionnement des Github Pages sur [la documentation officielle](https://help.github.com/articles/what-is-github-pages/).

Pour mettre en place un hébergement de type GitHub Pages, procédez aux actions suivantes :

1. Créez un repository GitHub intitulé de la forme « username.github.io » (où « username » correspond à votre login GitHub

/!\ Attention, si vous ne respectez pas précisément cette syntaxe, cela ne fonctionnera pas.

1. Téléchargez et installez [GitHub Desktop](https://desktop.github.com/) (mais vous pouvez très bien le faire en ligne de commande)
2. Clonez votre repository via le menu « File > Clone Repository » en y précisant cette adresse : <https://github.com/username/username.github.io>
3. Sauvez votre projet

Ça y est vous disposez d’un espace de stockage pour votre blog.

Pour le tester, rien de plus simple :

1. Créez avec un éditeur (dans mon cas [Visual Studio Code](<https://code.visualstudio.com/)>) un fichier « index.html »

/!\ Par défaut, sous Windows, votre repository sera situé ici : C:\Users\username\Documents\GitHub\username.github.io

Vous pouvez donc enregistrer ce fichier sous ce chemin.

1. Inscrire dans le fichier le contenu suivant :

|  |
| --- |
| <!DOCTYPE html>  <html>  <body>  <h1>Mon titre</h1>  <p>Mon blog sera là.</p>  </body>  </html> |

1. Le fichier apparaît alors sous GitHub Desktop dans la partie « changed files ». Vous pouvez entrer un commentaire de commit et cliquer sur « Commit to master » puis « Fetch origin ».

/!\ Il va falloir que vous patientiez quelques dizaines de secondes pour voir apparaître votre travail sur Internet.

1. Ouvrez un navigateur Web et entrez l’adresse suivante : https://username.github.io.

Ça y est, vous avez votre site disponible sur Internet ! Vous pouvez retrouver l’ensemble de ces étapes détaillez [ici](<https://pages.github.com/>).

Passons maintenant à l’étape suivante : intégrer un site avec Jekyll.

## ## 2ème étape : Bloguer avec Jekyll

[Jekyll]( <https://jekyllrb.com/>) est une façon de créer votre site statique sous le format que j’appellerais « blog ». Grâce à ce système, vous pouvez écrire ainsi vos billets au format Markdown et les intégrer rapidement.

Pourquoi utiliser Jekyll plutôt qu’autre chose ? J’utilise Jekyll pour plusieurs raisons :

La raison essentielle est que GitHub le met en avant. La compatibilité Jekyll est donc assurée au sein d’un hébergement de type GitHub Pages.

* Format d’article Markdown : un langage de balisage permettant la réutilisation des articles sous
* Simplicité de mise en place et de mise à jour

Vous trouverez plus d’information sur [la documentation officielle]( https://jekyllrb.com/docs/).

Pour le mettre en place, suivez les actions suivantes :

1. Installer un environnement de développement Ruby

Pas de panique, pour l’installer, il vous suffit de suivre ce qui est précisé dans [ce guide d’installation sous Windows](<https://jekyllrb.com/docs/installation/windows/>).

* 1. Téléchargez « Ruby+Devkit » et suivez l’installation pas à pas

/!\ Dans mon cas, je me base sur l’installeur « Ruby+Devkit 2.5.3.1 (x64) » qui est la version recommandée du moment.

* 1. Après installation, une invite de commande va se lancer, sélectionnez le mode d’installation numéro 1. Celui-ci vous permettra ensuite d’utiliser la commande « gem ».

1. A la fin de l’installation, sélectionnez « Start Command Prompt with Ruby » dans le menu des applications Windows. Une invite de commande se lance.
2. Procédez à l’installation de Jekyll avec la ligne de commande suivante :

|  |
| --- |
| gem install jekyll bundler |

1. A la fin de l’installation, vérifez sa bonne installation avec la ligne de commande :

|  |
| --- |
| jekyll -v |

Jekyll est maintenant installé sur votre poste, il ne vous reste plus qu’à l’utiliser.

Passons maintenant à l’étape suivante : réaliser son premier site avec Jekyll.

## ## 3ème étape : Réaliser son premier site avec Jekyll

Maintenant que Jekyll est installé, nous allons créer notre premier site en quelques commandes. Je ne vais donc pas être exhaustif et parcourir volontairement les grandes lignes. Le reste pourra être étudié plus en profondeur au sein de [la documentation officielle]( https://jekyllrb.com/docs/).

J’ai décidé volontairement de ne pas créer le site directement au sein de la copie locale de mon repository GitHub. Je réalise donc ensuite des copies locales lorsque j’obtiens une version que je juge satisfaisante de mon site.

Je vais donc vous décrire ce mode opératoire :

1. Ouvrez une invite de commande et se placer dans le répertoire racine où vous souhaitez y déposer votre site localement :

|  |
| --- |
| cd C:\Developpement |

1. Créez un nouveau site Jekyll préconfiguré :

|  |
| --- |
| jekyll new monsite |

/!\ Vous disposez désormais d’un répertoire « monsite » au sein de votre répertoire racine. Celui-ci contient votre site statique.

1. Déplacez vous dans le répertoire de votre nouveau site :

|  |
| --- |
| cd monsite |

1. Construisez le site et déployez le localement :

|  |
| --- |
| bundle exec jekyll serve |

1. Ouvrez un navigateur Web et entrez l’adresse suivante : https://localhost:4000/

Vous avez votre site Jekyll en local. Vous pouvez retrouver l’ensemble de ces étapes détaillez [ici]( <https://jekyllrb.com/docs/>).

1. Pour le mettre sur votre hébergement GitHub Pages, il vous suffit de copier le contenu de votre répertoire monsite/ et de le coller dans votre repository local « username.github.io ».
2. Les fichiers apparaissent alors sous GitHub Desktop dans la partie « changed files ». Vous pouvez entrer un commentaire de commit et cliquer sur « Commit to master » puis « Fetch origin ».

Ça y est, vous avez maintenant un site fonctionnel accessible sur votre espace GitHub Pages.

Passons maintenant à l’étape suivante : configurer son site Jekyll.

## ## 4ème étape : Le b.a.-ba pour configurer son site Jekyll

Vous disposez désormais d’un site Jekyll et vous voulez donc le configurer et le customiser. Je vais vous donner ici quelques pistes pour le faire. Je découperais mon explication en 3 thématiques :

* Comment configurer votre site ?
* Comment ajouter des posts ?
* Redéployer son site

Vous trouverez plus d’information sur [la documentation officielle]( https://jekyllrb.com/docs/).

### ### Comment configurer votre site ?

A la racine de votre site, vous trouverez un fichier « \_config.yml ». Vous pouvez librement y éditer, entre autres :

* Le titre de votre site via la balise « title »
* La description de votre site via la balise « description »
* Un lien vers votre compte GitHub avec la balise « github\_username »
* Etant une page du site et non un post, ce fichier à une disposition (layout) de type « page »

Vous pouvez modifier votre page « A propos » par l’intermédiaire du fichier « about.md ». Celui-ci est au format Markdown.

/!\ Un conseil, pour vous familiariser avec la syntaxe Markdown, le mieux est d’essayer de customiser et d’en voir les effets.

Vous pouvez accéder à la documentation sur la syntaxe et le principe des pages Jekyll [ici](https://jekyllrb.com/docs/pages/).

### ### Comment ajouter des posts ?

Comment vous pouvez le voir à l’initialisation de votre site, celui-ci dispose d’un post s’intitulant « Welcome to Jekyll ». Celui-ci se situe dans le répertoire « \_posts » et se trouve dans le fichier « AAAA-MM-dd-welcome-to-jekyll.markdown ». Il se compose, entre autres, des éléments suivants :

* Un titre de post via la balise « title »
* Une date de parution via la balise « date »
* Un détail du post situé après la ligne « --- »
* Etant un post du site et non une page à proprement parlé, ce fichier à une disposition (layout) de type « post »

Chaque ajout d’un nouveau fichier Markdown au sein du répertoire « \_post » créera un nouveau post au sein de votre site.

/!\ Un conseil, pour vous familiariser avec la syntaxe Markdown, le mieux est d’essayer de customiser et d’en voir les effets.

Vous pouvez accéder à la documentation sur la syntaxe et le principe des posts Jekyll [ici](https://jekyllrb.com/docs/posts/).

### ### Déployer son site

Après avoir réalisé des modifications, il vous est nécessaire de regénérer votre site Jekyll. La gestion de votre site Jekyll passe par quelques lignes de commande (mais rien de très sorcier).

Deux commandes sont à retenir :

|  |
| --- |
| jekyll serve |

Utile pour construire votre site à partir du moment où l’un de vos fichiers sources à été modifié. Le site est alors visible localement à l’adresse suivante : <https://localhost:4000/>

|  |
| --- |
| jekyll build |

Utile pour générer votre site pour l’environnement de Production. Typiquement, lorsque vous pensez disposer d’une version satisfaisante à mettre en ligne, vous pouvez la lancer puis récupérer le contenu du dossier « \_site » et le déployer sur votre environnement GitHub Pages.

Le reste pourra être étudié plus en profondeur au sein du [référentiel des commandes utiles]( https://jekyllrb.com/docs/usage/)

## ## Conclusion

Vous avez appris grâce à ce billet à configurer votre environnement GitHub Pages, y créer un site grâce à Jekyll et découvert les quelques notions de base pour le configurer.

Cela n’exclue évidemment pas un tour sur la documentation officielle pour aller plus loin dans votre utilisation de Jekyll. Néanmoins, cela vous rend autonome à a mise en place d’un site basique mais modulaire et tout ça très rapidement.