
titre: Créer son blog ou son site internet

auteur: subversive

date: 11-12-2019

Ouvrir un site internet sur le domaine public vous engage à respecter les lois en vigueur sur le territoire de votre résidence. Si vous ne souhaitez pas subir le droit en vigueur, vous pouvez vous orienter vers un site caché, .onion.

J'ai utilisé cette méthode, je souhaitai donc vous la partager.

Elle n'est pas recommandable pour un site à but lucratif car il s'agit ici d'un CMS statique.

Bien évidemment, la création d'un site n'est pas une ligne parfaite, vous devrez affronter des inconvénients techniques.

Préparation

[7-zip](#) - Traitement de dossier -

[Hugo](#) - CMS -

Un compte [Github](#) - Votre dépôt internet -

Un peu de place dans vos documents - Votre dépôt local -

[Git](#) - Gestion de dépôt -

Un [thème hugo](#) -

Un éditeur de code, dans l'exemple nous prendrons [sublime text](#) -

[7-zip](#)

Créez votre compte, téléchargez les applications, choisissez votre thème (Il est possible d'en avoir plusieurs bien sur).

Installation

Installez 7-zip.

Installez Git.

Sur Github, téléchargez la dernière release Hugo pour votre OS, dans notre exemple, on prendra 60.1-Windows-64bit.

Extraire le dossier.

Dans vos documents créez un dossier que vous nommez **Hugo** (respectez la majuscule).

Dans ce dossier, créez un dossier **bin**, copiez-y les fichiers extraits de la release Hugo.

Allez dans le dossier bin, en haut dans l'adresse locale, tapez cmd, puis entrée. Une fois dans l'invite de commande(cmd) collez-y :

```
set PATH=%PATH%;C:\Hugo\bin
```

Fermez l'onglet cmd, allez dans le dossier Hugo, ouvrez-y cmd.

```
hugo new site Blog
```

Votre Site est créé. Il est dans le fichier Blog.

Design

Nous allons utiliser pour l'exemple un seul thème, [Hermit](#).

Pour plus de souplesse, nous n'allons pas suivre la procédure recommandé par le créateur du thème.

Cliquez sur download, puis sur github téléchargez-le :

Faites extraire. Allez dans les dossiers du thème. Chaque thème est unique. Afin de pouvoir le modifier, ainsi que l'optimiser à ses goûts, nous allons procéder à des petites modifications. Créez un dossier `theme1`, puis y insérez les dossiers et fichiers du thème :

- * archetypes
- * assets
- * i18n
- * layouts
- * .editorconfig
- * .gitattributes

Allez dans le dossier de votre site hugo/Blog, lancez le cmd. Dicter les commandes :

```
hugo new about.md
hugo new posts/article1.md
```

Copiez votre dossier `theme1`, puis collez le dans Hugo/Blog/thèmes/ .

Copiez le dossier static du thème et collez-le dans votre Blog. Faites de-même pour le dossier resources.

Egalement pour le fichier `config.toml` dans examplesite. Remplacez-le. Ouvrez-le avec votre éditeur de code.

Nous allons le configurer :

Allez dans archetypes/default.md, mettez `draft : false`. Faites de même dans `theme1/archetypes/default.md` et `posts.md` .

Laissez vide la baseUrl pour l'instant. Nommez le titre de votre Blog. Nommez `theme1` pour votre thème. Mettez le en français `fr`.

Config :

Bien votre site tourne ! Testez le via cmd depuis Hugo/Blog, tapez `hugo server -D`, rendez vous dans votre navigateur a l'adresse : `http://localhost:1313/`.

Maintenant peaufinez votre site, via `config.toml`, et dans votre thème dossier layouts. Si vous avez plusieurs thèmes, il est très recommandable d'avoir un fichier config pour chaque thème et de le changer à chaque changement de thème. Dans config, le # permet de désactiver les commandes.

On ne va pas rentrer dans les détails chaque thème est différent.

Pour écrire des articles, des pages, vous utiliserez le cmd et la commande `hugo new` pour créer une page ou `hugo new posts/nomdupost.md`, ensuite utilisez un éditeur de code pour enregistrer vos modifications. Laissez tourner le serveur hugo pendant la réalisation de cette étape car à chaque enregistrement, il se met à jour très rapidement.

Déploiement

Dans cette étape, nous allons utiliser une technique peu coûteuse, vous pouvez si vous le souhaitez ne pas payer. Nous utiliserons Github.

Avant toute chose, dans le dossier de votre blog, créez un fichier `netlify.toml` et copiez-y :

```
[build]
publish = "public"
command = "hugo"
```

```
[context.production.environment]
HUGO_VERSION = "0.60.1"
HUGO_ENV = "production"
HUGO_ENABLEGITINFO = "true"
```

```
[context.branch-deploy.environment]
HUGO_VERSION = "0.60.1"
```

```
[context.deploy-preview.environment]
HUGO_VERSION = "0.60.1"
```

```
[context.deploy-preview]
command = "hugo -b $DEPLOY_PRIME_URL --buildFuture"
```

```
[context.branch-deploy]
command = "hugo -b $DEPLOY_PRIME_URL --buildFuture"
```

Changez la version de hugo par celle que vous utilisez.

Clé SSH :

Allez dans Github settings sous votre profil → SSH and GPG → New SSH key. Donnez y un nom.
Retourner dans le dossier de votre site → clic droit → git bash here.
Commande :

```
ssh-keygen -t rsa -b 4096 -C "votremail@mail.com"
```

```
clip < ~/.ssh/id_rsa.pub
```

Retournez sur Github collez votre clé. Validez.

Sur Github créez un nouveau projet Blog. Vous arrivez ensuite sur la page du dépôt qui est vide.
Dans le dossier de votre Blog, clic droit, Git Bash Here. Stoppez l'activité en local de votre site.

Poursuivons sur votre dépôt :

PS: Pour coller du texte dans Git, clic droit paste.

Hugo/Blog → clic droit → git bash here.
Commandes :

```
git init
git remote add origin git@github.com:moncompte/Blog.git
git add .
git commit -m "Premier envoi"
git push -u origin master
```

Votre Dépôt est à jour. Continuons. Allez sur [Netlify](#). → Get Started for free. Connectez-vous via Github.

Puis New site from Github. Sélectionnez votre compte Github. Puis le dépôt de votre Blog. Déployez-le.

Le site déployé dans notre exemple fonctionne. Votre site est normalement sur le net.

Intégration

La première chose à faire est de renommer votre domaine. Allez dans site settings → change site name. Vous vous retrouvez avec exemple-blog.netlify.com . Changez votre baseUrl dans config, en local comme sur Github.

Pour ceux ne souhaitant pas payer, la suite n'est pas pour vous.

Si vous ne possédez pas de nom de domaine, achetez-en un. Désactivez le protocole DNSSEC. Enregistrez quatre Serveurs DNS externes :

```
dns1.p02.nsone.net  
dns2.p02.nsone.net  
dns3.p02.nsone.net  
dns4.p02.nsone.net
```

Retournez sur la page overview de votre site netlify → Domain settings → add custom domain → collez le vôtre. Forcez et/ou attendez la mise en route HTTPS.

Nous allons tenter une redirection de votre domaine netlify vers votre domaine externe, pour cela rendez-vous dans le dossier static de votre site sur le dépôt github :

Ouvrez le EDIWeb, créer un fichier `_redirects` Copiez-y :

```
# Redirect default Netlify subdomain to primary domain  
https://votresite.netlify.com/* https://votredomaine.fr/:splat 301!  
http://votresite.netlify.com/* https://votredomaine.fr/:splat 301!
```

Enregistrer dans la branche master.

Penser à mettre le dossier sur votre disque local au même endroit (static).

Voilà votre site tourne !

A vous de jouer pour obtenir du trafic ! Chacun sa méthode.

Dernière chose pour ne pas vous laisser dans l'embarras.

Mettre à jour son site

Assez complexe, tout d'abords ne changez jamais le nom du dépôt Github. Ainsi que le local. Exercez votre mise à jour avec `hugo server -D` désactivé via le cmd.

Puis clic droit → GitBash here. Voici la suite de commande :

```
git status  
git add .  
git commit -m "nomdelamiseajour"  
git push -u origin master ou git push -f origin master
```

Normalement votre site se met à jour directement via le dépôt Github.

Pensez à vérifier de temps en temps sur netlify, que les mises à jour soient bien publiées, il peut-y avoir des surprises !