

projet git mypelican_static_sites

Vue d'ensemble

Il s'agit d'un projet github qui utilise un générateur de sites appelé pelican pour générer des sites web statiques.

Au moment de la rédaction, ce projet maintient le contenu et la génération pour trois sites distincts. Chaque site est rendu en 3 langues : anglais, français et espagnol.

Par convention, nous nommons chacun de ces sous-répertoires en utilisant le nom de domaine du site. Les scripts s'appuient sur cette convention.

Sites actuellement maintenus

1. www.bernatchez.net
Hébergé sur Bunny.net : www-bernatchez-net
2. www.ogopogo.biz
Hébergé sur Bunny.net : www-ogopogo-biz
3. blog.bernatchez.net
Hébergé sur Bunny.net : blog-bernatchez-net
4. docs.pelican.bernatchez.net
Hébergé non publié sur le système de fichiers local.

Génération du site web

Générez une version locale d'un site afin de pouvoir le parcourir et le vérifier avant de le publier.

```
cd ./blog.bernatchez.net
make html
xdg-open ./bucket/index.html
cd ..
```

Générez une version en ligne du site et publiez-la.

```
cd ./blog.bernatchez.net
# Assurez-vous que le fichier deploy.bunny exécute réellement une commande "nmx" plutôt que de simplement l'afficher.
make pubhtml

# Invoquez manuellement la commande affichée en dernière ligne par le make ci-dessus
# Cela pousse tout vers le cache CDN
npx --yes bunny-transfer@latest purge blog-bernatchez-net-zone
```

Avertissement

Une fois que vous avez invoqué « make pubhtml », le contenu de votre « site local » ne sera plus approprié pour une consultation locale. Si vous souhaitez parcourir les versions locale et en ligne de façon interchangeable, vous devez répéter la commande « make html » après la publication, laissant ainsi les fichiers locaux configurés pour une consultation locale.

Maintenance du site web

Portée

Le projet git est organisé avec 2 portées pour le contenu source du site.

Portée source globale

Les fichiers .rst contenus dans les sous-répertoires du répertoire principal du projet. Ces fichiers peuvent être utilisés comme source par n'importe lequel de nos sites web.

Portée source locale

Les fichiers .rst contenus dans les sous-répertoires d'un sous-répertoire de site donné. Ces fichiers ne peuvent être utilisés comme source que par le site web concerné.

Le sous-répertoire « content » d'un site donné contient des sous-répertoires dont les noms correspondent aux sous-répertoires de portée globale ou locale. Cela détermine ce qui sera inclus dans le site.

Les noms de répertoires présents dans le répertoire « content » déterminent ce qui est inclus dans le site généré.

Le fichier make se charge de peupler ces répertoires avec des liens symboliques vers les fichiers source .rst dans leurs répertoires de portée locale ou globale correspondants.

En ce qui concerne le dépôt, vous ne faites « git add » que pour les fichiers .rst situés dans les répertoires de portée globale ou locale. Les sous-répertoires sous content/ ne contiendront qu'un fichier « placeholder.txt » dans git pour s'assurer que leurs sous-répertoires seront créés lors du clonage.

Source pdf

Tous nos sites contiennent également des versions .pdf de leurs pages. Pour produire ces fichiers pdf, chaque répertoire source contient deux fichiers de script exécutables : **article2pdf** et **qarticle2pdf**. Ce dernier est simplement une version moins verbeuse. Chaque fois que vous modifiez ou ajoutez du contenu dans un répertoire source, pensez à régénérer les fichiers .pdf en conséquence.

En général, vous modifiez votre contenu, puis invoquez « qarticle2pdf » pour vous assurer que vos pdfs sont à jour.

Si vous avez créé un nouveau répertoire source, vous pouvez générer ces deux scripts en invoquant la commande « genarticle2pdf -q » à l'intérieur du nouveau répertoire.

Maintenance github

Lorsque vous clonez le projet github dans un répertoire local, vous obtenez un répertoire principal qui contient des sous-répertoires de sites web, chacun d'entre eux étant un répertoire de génération de site web indépendant.

Le répertoire principal du projet contient divers sous-répertoires de *portée source globale* (ex. : *./examples*, *./artwork*, *./travel*, etc.). Chacun d'eux contient des fichiers .rst pouvant servir de source pour la génération du contenu du site. Tous sous contrôle de révision git.

Ce contenu de *portée source globale* peut optionnellement être utilisé comme source pour n'importe lequel des sites web.

Chaque sous-répertoire de site web peut également contenir ses propres sous-répertoires contenant des fichiers .rst de *portée source locale* spécifiques au site (ex. : *./blog.bernatchez.net/pages/*) destinés uniquement au site concerné.

Pour ajouter l'un de ces répertoires de contenu à *portée locale ou globale* à un site web donné, vous devez créer un répertoire du même nom sous le sous-répertoire content du site concerné. Vous devez également ajouter un fichier « placeholder.txt » et le mettre sous contrôle de révision (ex. : `git add placeholder.txt`) pour s'assurer que ce nouveau répertoire sera créé lors de l'utilisation de « git clone ».

Par exemple, nous avons procédé ainsi pour ajouter *./examples* au site web *./blog.bernatchez.net* :

```
mkdir ./blog.bernatchez.net/content/examples/  
cp ./blog.bernatchez.net/content/pages/placeholder.txt ./blog.bernatchez.net/content/examples/  
git add ./blog.bernatchez.net/content/examples/placeholder.txt
```

Vous devez également modifier le fichier `.gitignore` du projet pour éviter d'ajouter les liens symboliques générés sous `content/examples` dans le dépôt git. Ces liens sont régénérés à chaque `make`.

Nous avons ajouté les lignes suivantes au fichier `.gitignore` :

```
blog.bernatchez.net/content/examples/  
blog.bernatchez.net/content/examples/*
```

Dépôt

La copie centrale originale du dépôt de ce projet est :

[git@github.com:pierrebernatchez/mypelican_static_sites.git](https://github.com/pierrebernatchez/mypelican_static_sites.git)