
titre: Ecrire des commentaires sur un site statique (hugo) avec staticman et heroku

auteur: subversive.eu

date: 02-05-2020

Bonjour !

Tutoriel pour permettre l'écriture de commentaires sur un site internet statique avec [hugo cms](#), [staticman](#) et [heroku](#).

Objectif, offrir a vos visiteurs la possibilité d'écrire des commentaires (ou autres) sans passer par un organisme tel que Graphcomment ou Disqus,... Ainsi vous respectez la vie privée de vos utilisateurs.

Tous les tutos disponibles sur le net (que j'ai lu) sont en anglais, ils datent et certains liens ne fonctionnent plus, bref j'ai du faire un mix de tuto pour y arriver, du coup je vous propose un tuto en français.

N'oubliez jamais que ce qui fonctionne est sur les codes sources des sites. En ce sens si vous copiez-collez et qu'il y a une erreur, cherchez la mise à jour dans le code source sur github des sites qui tournent avec.

Requis

- Votre compte github principal.
- Un compte github vérifié qui sera votre bot (ex: githubbot). Conservez les identifiants et mots de passe.
- Un compte vérifié [heroku](#). Conservez les identifiants et mot de passe.
- Heroku CLI.

J'utilise [heroku pour ubuntu 20.04](#), les commandes sont un petit peu différentes mais la procédure reste la même.

Considérant que vous souhaitez installer staticman sur votre site, vous avez donc la capacité de gérer correctement votre site sous hugo (ou autre), de traduire les .toml en .yaml et vice versa.

Bien il vous faudra de la patience pour ne manquer aucune étape.

Commençons !

Compte githubBOT

Connectez vous à votre compte GithubBOT, allez dans paramètres du compte (settings du compte) -> Developer settings -> Personal access tokens -> Generate new token.

Nommez le, sélectionnez (select scopes) : repo (tous) + user (tous) => Generate token. SAUVEGARDEZ bien votre token sur un fichier txt.

Application Heroku

[Télécharger staticman via github](#) avec le lien https :

```
git clone https://github.com/eduardoboucas/staticman.git
```

Allez dans le dossier staticman.

Créez un fichier, nommez-le "Procfile", insérez-y "npm start".

Créez une clé RSA :

```
openssl genrsa -out key.pem
```

Créez un fichier, nommez-le "config.production.json", insérez-y :

* notes :

> votretoken ressemble a celui ci : b81f780f9f5f3be15a5595e618379756344cdsdk2
> pour la rsaPrivateKey, collez l'intégralité du fichier key.pem

```
{  
  "githubToken": "votretoken",  
  "rsaPrivateKey": "-----BEGIN RSA PRIVATE KEY\n-----la clé en entier-----\nEND RSA  
PRIVATE KEY-----",  
  "port": 8080  
}
```

(toujours dans le dossier staticman)

Connectez votre compte heroku avec :

```
heroku login
```

Toujours dans le dossier cloné plus haut (staticman), créez une application heroku :

```
heroku create nomdevotreapplication
```

Configuration

(votretoken ressemble a celui ci : b81f780f9f5f3be15a5595e618379756344cdsdk2)

```
heroku config:set NODE_ENV="production"  
heroku config:set GITHUB_TOKEN="votretoken"  
heroku config:set RSA_PRIVATE_KEY="$(cat ./key.pem)"
```

Créez une branche de production :

```
git checkout -b production 55d1430
```

Ajoutez dans le fichier .gitignore pour ignorer le fichier configuration:
"!config.production.json"

Déploiement

```
git add .  
git commit -m "Set up Staticman v3 for deployment to Heroku"  
git push heroku production:master
```

Si tout c'est bien passé, en se rendant sur
<<https://nomdevotreapplication.herokuapp.com/>> vous devez lire :

Hello from Staticman version 3.0.0!

Votre app est disponible.

API sur github

Connectons votre githubBOT avec votre compteGITHUBprincipal.

Pour cela, depuis le dépôt de votre site, allez dans "settings" (paramètres de ce dépôt)
-> Manage Access -> Invite a collaborator -> yourgithubBOT -> Invite.

Acceptez l'invitation via le mail de votre githubBOT.

API ponts et config

Depuis le fichier staticman.yml, j'ai choisi volontairement de désactiver la modération, mais faites à votre guise.

```
moderation: true
```

Cela va créer une "pull request" sur votre branche sélectionnée dans staticman.yml.

A la source (la RACINE) de votre site, créez un fichier "staticman.yml" , insérez-y [ce qu'il y a dans le liens](#).

Dans votre fichier config.yml, ajoutez ou complétez :

```
params:  
  comments: true  
  staticman:  
    api:  
      'https://nomdevotreapp.herokuapp.com/v2/entry/githubprincipal/repositorydusite/  
master/comments'
```

Ajoutez également un dossier `/data/` puis `/comments/` et un fichier .gitkeep.
(Cela permet de maintenir le git push avec les dossiers. Essayez sans, les dossiers vides ne suivront pas sur github).

La variable "comments" vous permet de désactiver les commentaires depuis le fichier `single.html`, vous pouvez aussi l'utiliser dans les params d'un article (ou page : `comments: false`).

Afficher et écrire des commentaires depuis le site

default

Dans `layout/_default/single.html` insérez :

```
{{- if and ($.Site.Params.comments) (ne .Params.comments false) }}  
{{- partial "comments.html" . }}  
{{- end }}
```

[fichier single.html](#). (cf juste au-dessus pour activer/désactiver)

html

Dans `layouts/partials/comments.html` (créez le fichier si besoin "comments.html") insérez-y [partials comments.html](#).

javascript

Le javascript est nécessaire dans cette config, à vous de le personnaliser en fonction de vos désirs.

Pour des soucis de performance, le script se trouve dans le fichier [script.js](#).

> Notes: Dans [le partial footer](#), mettez votre script en bas des pages pour éviter les erreurs.

css

Pour styliser le formulaire et l'affichage avec les boutons c'est ici : [CSS styles](#).

A vous de choisir en fonction de votre architecture css, en fonction du thème choisi.

Pour styliser le captcha c'est plus complexe cherchez `// Captcha SETUP Initial` dans [script.js](#).

Protection contre le spam

A la recherche de l'utilité, de la vie privée et de la performance.
[Buster: Captcha Solver for Humans](#).

Les CAPTCHA diminuent les performances de vos sites web /apps.

Le REcaptcha favorise a outrance chromium et les comptes google.

Il faut que la réponse soit sur un serveur distant, sinon le robot (peut) lire la réponse ?
En attente de confirmation.

Inutile si peu de visiteurs, car le spam est un marché, qui va ouvrir un commerce où il n'y a personne ?

J'utilise 3 protections (toutes dépassées):

- Pas de javascript = pas de commentaire. Il faut javascript pour afficher les commentaires car il faut cliquer sur le bouton "ouvrir espace commentaire".
- Un input caché `<input type="hidden" id="youarenotarobot1">`, le visiteur(non-robot/humain) ne le voit pas mais le robot oui, si il le remplit alors ce n'est pas un humain.
- [js-captcha de robiveli](#).

autre solution

Posséder un serveur, y ajouter staticman + [visual captcha](#).

Vous pouvez également interdire les url dans le contenu des commentaires.

Notes importantes

- Il n'y pas de protection contre le spam vraiment active, si jamais lisez l'article [Tuto en anglais de 2020](#).
- Il n'y a pas de modération avant affichage.
- Ne pas hésiter à observer [le code source du site](#).
- Cette version est très simple, sans réponse entre commentaires, avec trois champs requis seulement.
- Pour observer [où se stocke un commentaire](#).

Astuce

* Je maîtrise mal le vocabulaire git mais avant de push -u vers le dépôt github, pensez à utiliser "git pull" pour "rapatrier" les nouveaux commentaires sur votre dépôt local.

* Dans la version (v3) de staticman, il n'y aurait plus besoin du Personal access tokens. Mais je ne sais comment procéder.

Problème(s) rencontré(s)

- L'utilisation de `{{ $slug := replace $.RelPermalink "/" "-" }}` avec `name="options[slug]"` ne permet l'affichage des commentaires sur les articles où les titres ont des accents. Solution trouvée avec `{{ replace $.File.Path "/" "-" }}`.
- J'ai supprimé du script de js-captcha, je ne sais à quoi il sert, il se peut qu'il ne fallait pas le faire.
- De même pour staticman, le script ReCaptcha n'est pas dispo sur les fichiers de ce site.
- J'ai un souci de cache-control depuis que les commentaires sont disponibles [résolu].
- Il faut reconstruire le site pour afficher un commentaire, ce qui peu prendre du temps, surtout si le dino (herokuapp) dort..

Sources

tutos

[Tuto en anglais de 2018](#)

[Tuto en anglais de 2019](#)

[Tuto en anglais de 2020](#)

Tuto en anglais de 2020 2

fichiers

Ceux au complet dont je me suis servi.

[staticman-js-common.js](#)

[staticman-styles.css](#)

[staticman.html](#)

[js-captcha-index.js](#)

[js-captcha-index.html](#)