

## Le référencement de sites web dynamiques (3ème partie)

[Retour au sommaire de la lettre](#)

*Le référencement de sites web dynamiques est l'une des principales sources d'interrogations des webmasters actuels. Après avoir longtemps été un facteur totalement bloquant pour les moteurs de recherche, la situation s'est assouplie depuis quelques temps. Le mois dernier, nous avons étudié quels sont les types de pages qui sont compatibles avec les moteurs et ceux qui ne le sont pas, ainsi que les problèmes posés aux moteurs par les sites dynamiques. Ce mois-ci, nous nous penchons avec Olivier Duffez (du site WebRankInfo) sur la technique de l'URL rewriting, qui constitue certainement la meilleure solution pour obtenir un meilleur référencement des sites web dynamiques.*

Nous avons vu, les mois précédents dans cette lettre, que les sites web dynamiques sont parfois des obstacles pour les moteurs de recherche, mais que la situation a plutôt tendance à s'améliorer. Cependant, parfois, certaines pages représentent un réel obstacle pour les robots et il faut intervenir, mettre en œuvre une procédure spécifique, pour faire en sorte qu'un site soit indexé par les « Google et consort ». Parmi les sept solutions envisagées le mois dernier, celle de l'URL rewriting (ou « réécriture d'URL ») semble la meilleure et, en tout cas, la plus efficace. Pour aborder de façon plus approfondie cette technique, nous avons demandé à Olivier Duffez de nous décrire la façon de mettre en place des fonctions d'URL Rewriting sur un site web. Une tâche qu'il connaît bien puisqu'il l'a mise en place sur son site WebRankInfo (<http://www.webrankinfo.com/>). Laissons-lui la parole, ou plutôt la plume (voire le clavier ;-))...

**Nota :** cet article est inspiré du contenu du chapitre « Le référencement de sites web dynamiques » du livre « Google, trucs de pros » à paraître chez Micro Application (<http://www.livre-google.com/>). Son contenu cependant été revu pour l'adapter à cette lettre.

Comme il a été indiqué dans les articles précédents de cette lettre professionnelle, les sites dynamiques génèrent la plupart du temps des pages aux URLs longues et complexes, du fait de la présence de variables. Avec des pages dynamiques, un seul fichier permet de créer dynamiquement des centaines ou des milliers de pages. Une fois que la réécriture d'URL est mise en place sur un site, les pages seront accessibles grâce à des URL « propres », que ce soit pour les internautes ou pour les robots des moteurs de recherche.

Ainsi, une page qui était accessible à l'adresse :

**<http://www.notre-site.com/articles/article.php?id=12&page=2&rubrique=5>**

sera accessible après URL rewriting à l'adresse (par exemple) :

**<http://www.notre-site.com/articles/article-12-2-5.html>**

Ces URL dites « propres » (débarassées de caractères spéciaux comme « ? » ou « & ») facilitent l'indexation des sites dynamiques, et donc leur référencement dans les moteurs.

En plus de cet avantage indéniable, la réécriture d'URL permet également de renforcer la sécurité du site en masquant les noms des variables passées dans l'URL. Si l'extension des URL « propres » est neutre (par exemple .html ou .htm), il est même possible de masquer le langage utilisé sur le serveur (PHP dans notre exemple).

### Principe de l'URL Rewriting

Le principe de la réécriture d'URL est donc de mettre en place un « système » sur le serveur pour qu'il sache interpréter ce nouveau format d'URL. Dans notre exemple, quand un visiteur accède à la page **<http://www.notre-site.com/articles/article-12-2-5.html>**, le serveur doit renvoyer exactement la même chose que si le visiteur avait demandé à accéder à la page **<http://www.notre-site.com/articles/article.php?id=12&page=2&rubrique=5>**.

La correspondance entre les deux schémas d'URL est alors décrite sous forme de « règles de réécriture ». Chaque règle permet de décrire un format d'URL. Dans l'exemple ci-dessus, la règle de réécriture va indiquer au serveur de prendre le *premier nombre* comme *numéro d'article*, le *deuxième* comme *numéro de page* et le *troisième* comme *numéro de rubrique*.

La technique de réécriture d'URL la plus connue est celle disponible sur les serveurs Apache, le plus souvent utilisés avec le langage PHP. Sauf mention spéciale, tous les exemples de cet article seront donc consacrés au langage PHP et au serveur Apache.

### **Mise en place de l'URL Rewriting**

Si vous avez déjà un site dynamique en ligne, voici les étapes à suivre pour mettre en place la réécriture d'URL :

1. Vérifier que votre hébergeur permet l'utilisation de l'URL Rewriting.
2. Identifier les pages dynamiques dont l'URL comporte des paramètres, et choisir un nouveau schéma d'URL « propre ».
3. Ecrire les règles de réécriture dans le fichier .htaccess adéquat.
4. Changer tous les liens vers chaque fichier dont l'URL a changé.
5. Mettre à jour votre site et vérifier que tout fonctionne.

Ouf... ;-) Voyons maintenant chacune de ces étapes les unes après les autres :

#### **1. Vérifier si l'URL Rewriting est compatible avec votre hébergeur**

La première chose à faire est bien évidemment de s'assurer que le serveur qui héberge votre site permet d'utiliser la réécriture d'URL. Tout dépend, dans un premier temps, du type de serveur utilisé. L'objet de cet article n'est pas de passer en revue tous les types de serveurs mais voici un résumé des possibilités de réécriture d'URL sur les serveurs web les plus courants :

<b>Serveur web</b>	<b>Support de la réécriture d'URL</b>	<b>Détails</b>
<b>Apache</b>	Géré par le module <i>mod_rewrite</i> , un module standard d'Apache à partir de la version 1.3.27	Le module <code>mod_rewrite</code> doit être actif : le fichier de configuration d'Apache ( <code>httpd.conf</code> ) doit contenir cette ligne : <b>LoadModule rewrite_module libexec/mod_rewrite.so</b> ainsi que celle-ci : <b>AddModule mod_rewrite.c</b>
<b>IIS (Microsoft)</b>	En ASP : réécriture possible par des filtres ISAPI, commercialisés par diverses sociétés (payants).	Le paramétrage des règles de réécriture est spécifique à chaque composant.

	En ASPX (.NET), sur tous les serveurs supportés : des fonctions sont disponibles comme RewriteURL(), etc. qui prennent en charge la réécriture d'URL.	Des codes prêts à compiler pour exploiter ces capacités sont fournis par Microsoft : <a href="http://www.microsoft.com/france/msdn/technologies/technos/asp/info/info.asp?mar=/france/msdn/technologies/technos/asp/info/2004-06-02-URLRewriting.html">http://www.microsoft.com/france/msdn/technologies/technos/asp/info/info.asp?mar=/france/msdn/technologies/technos/asp/info/2004-06-02-URLRewriting.html</a> ou via des projets open source comme : <a href="http://www.codeproject.com/aspnet/urlrewriter.asp">http://www.codeproject.com/aspnet/urlrewriter.asp</a> Aucune méthode standard n'a été prévue pour définir les règles de réécriture mais une utilisation pratique consiste à les paramétrer directement dans le web.config (fichier de configuration de l'application ASP.Net, présent notamment à la racine du site), qui est standardisé en XML.
--	---	---

Si votre site est hébergé sur un serveur dédié, vous avez accès vous-même à la configuration du serveur. Dans le cas d'un serveur Apache, vous pouvez donc modifier le fichier de configuration afin d'activer le support de la réécriture d'URL. Pensez à redémarrer Apache après avoir modifié le fichier de configuration.

Mais ce n'est pas tout. Si votre site est hébergé sur un serveur mutualisé, il n'est pas garanti que votre hébergeur ait activé le support de la réécriture d'URL, principalement pour des raisons de sécurité.

Si votre site est fourni par un hébergeur gratuit, il y a peu de chances que la réécriture d'URL soit possible. Nous vous conseillons fortement d'investir dans un hébergement payant (en plus d'un nom de domaine), les avantages sont réellement nombreux pour effectuer un bon référencement.

Pour vérifier si le module **mod\_rewrite** d'Apache est activé, il vous suffit de suivre les points suivants :

1. Créez un répertoire nommé test que vous placerez à la racine de votre site (il sera donc accessible via l'adresse **<http://www.votre-site.com/test/>**).

2. Dans ce répertoire, créez un fichier HTML nommé test.html (**<http://www.votre-site.com/test/test.html>**) contenant uniquement les lignes suivantes :

```
<html>
<head>
<title>Test</title>
</head>
<body>
OK !
</body>
</html>
```

3. Dans ce répertoire, créez un fichier nommé .htaccess contenant les lignes suivantes (nous verrons plus loin leur signification) :

```
Options +FollowSymLinks
RewriteEngine on
RewriteRule ^inconnu\.html$ /test/test.html [L]
```

4. Transférez ce répertoire et ces deux fichiers sur votre site, puis rendez-vous à l'adresse **<http://www.votre-site.com/test/inconnu.html>**

Vous vous attendez sans doute à ce que le navigateur affiche un message d'erreur indiquant que le fichier nommé inconnu.html n'existe pas à cet endroit sur votre site (erreur 404). Si c'est le cas, alors votre hébergeur n'autorise sans doute pas la réécriture d'URL : contactez-le pour lui demander.

Si vous êtes chez un hébergeur gratuit qui ne le gère pas, c'est une très bonne raison pour franchir le cap et bénéficier de tous les avantages d'un hébergement professionnel (qui est désormais financièrement accessible à tous).

Sinon, vous devriez voir le texte « OK ! », ce qui signifie qu'en demandant à voir le fichier inconnu.html (qui n'existe pas physiquement sur le serveur), le serveur vous affiche le contenu du fichier test.html (qui existe bien). C'est le principe même de la réécriture d'URL et donc la preuve que votre serveur gère bien la réécriture d'URL. Il a « réécrit », dans notre cas, « inconnu.html » en « test.html ». CQFD.

Il peut y avoir un troisième cas de figure (que nous ne vous souhaitons pas...) : votre site est totalement bloqué, aucune page ne peut être affichée, et vous avez un message indiquant « Erreur 500 ». Dans ce cas ne paniquez pas, il vous suffit de retirer le fichier .htaccess qui est incompatible avec votre hébergeur.

## 2. Définir les schémas d'URL

Reprenons notre exemple de site qui dispose d'une base de données d'articles, et dressons la liste des types d'URL. En voici quelques exemples :

- \* **article.php?id=12&rubrique=5**
- \* **article.php?id=12&page=2&rubrique=5**

Pour simplifier la lecture, nous n'avons listé ici que des URLs concernant le même article, mais dans la pratique quand vous essayez de dresser la liste des types d'URL sur votre site, vous pouvez tomber, par exemple, sur :

- \* **article.php?id=182&rubrique=15**
- \* **article.php?id=36&page=5&rubrique=3**

Le principe de l'URL rewriting consiste à trouver les schémas des URL à partir de leurs formes communes. Dans notre exemple, les articles sont accessibles selon deux types d'URL (*id+rubrique* ou *id+rubrique+page*), suivant que le numéro de page est précisé ou non.

A partir du moment où vous avez identifié ces « schémas d'URL », vous devez choisir un nouveau format d'URL (l'url « propre »). En général on fait apparaître un nom de fichier avec l'extension .html (ou .htm) mais sachez que vous pouvez mettre ce que vous voulez, cela n'a aucune incidence sur la prise en compte des pages par Google. En effet, quelle que soit l'extension que vous aurez choisie, la page restera une page respectant la norme HTML.

Le nom du fichier sera formé d'un préfixe et/ou d'un suffixe, et des valeurs des variables (que ce soient des chiffres ou des lettres). Profitez de cette étape pour bien réfléchir en fonction du référencement, car vous pouvez utiliser des mots-clés dans les URL de vos pages, qui soient plus parlants pour les internautes et sans doute pris en compte par les moteurs de recherche. Voici les nouveaux formats d'URL que nous avons choisis pour chacune des URL des exemples précédents :

- \* **article-12-5.html**
- \* **article-12-2-5.html**
- \* **article-182-15.html**
- \* **article-36-5-3.html**

Pour séparer les différentes parties de l'URL, vous devez choisir un séparateur (dans cet article, nous avons choisi d'utiliser uniquement des tirets). Il est plus efficace pour le référencement de choisir un caractère qui soit considéré comme un séparateur de mots par Google. Ainsi, vos URL pourront contenir des mots-clés, ce qui est pris en compte sans soucis par Google.

Vous pouvez néanmoins également utiliser les caractères suivants :

- Le tiret : -
- La virgule : ,
- Le point : .
- La barre oblique (slash) : /
- La barre verticale (pipe) : |

Nous vous déconseillons d'utiliser les caractères suivants :

- Le tiret bas (underscore) : \_

- Le signe dièse : #
- L'esperluette : &
- L'arobase : @
- Le point d'interrogation : ?
- Le signe dollar : \$
- Le signe dièse : #
- Les caractères accentués, l'espace

Le tiret et la virgule sont les plus simples ; la barre oblique peut poser des problèmes de répertoires et la barre verticale n'est pas très connue des internautes. Le tiret bas (underscore) pose des soucis avec Google.

Nous avons donc défini deux formats d'URL pour notre rubrique d'affichage des articles. Essayons de les formaliser en supprimant les numéros d'articles, de rubriques ou de pages, et en les remplaçant par leur signification :

- \* **article-ARTICLE-RUBRIQUE.html**
- \* **article-ARTICLE-PAGE-RUBRIQUE.html**

Bien entendu, ARTICLE, RUBRIQUE et PAGE représentent ici des numéros.

### 3. Rédiger les règles de réécriture

Maintenant que nous avons déterminé les différents schémas d'URL, il reste à écrire les règles de réécriture qui vont indiquer au serveur comment interpréter chacun de ces schémas.

Passons directement à la solution que nous allons commenter... Voici le contenu du fichier .htaccess situé dans notre répertoire <http://www.notre-site.com/articles/> :

```
#-----  
# Répertoire : /articles/  
#-----  
  
# Le serveur doit suivre les liens symboliques :  
Options +FollowSymlinks  
  
# Activation du module de réécriture d'URL :  
RewriteEngine on  
  
#-----  
# Règles de réécriture d'URL :  
#-----  
  
# Article sans numéro de page :  
RewriteRule ^article-([0-9+)-([0-9+)]\.html$ /articles/article.php?id=$1&rubrique=$2 [L]  
  
# Article avec numéro de page :  
RewriteRule ^article-([0-9+)-([0-9+)-([0-9+)]\.html$  
/articles/article.php?id=$1&page=$2&rubrique=$3 [L]
```

Note : il ne doit pas y avoir de retour chariot sur une ligne de règle de réécriture.

Les lignes commençant par le signe dièse (#) sont des commentaires. N'hésitez pas à en ajouter pour rendre vos fichiers plus compréhensibles : ces lignes sont totalement ignorées par le module de réécriture d'URL.

Chaque fichier .htaccess est spécifique à un répertoire ; nous avons pris l'habitude d'indiquer en haut de ce fichier l'emplacement du répertoire sur le site. Chaque répertoire de votre site devra donc proposer son propre fichier .htaccess.

Les deux premières instructions (**Options +FollowSymlinks** et **RewriteEngine on**) ne doivent être présentes qu'une seule fois par fichier, avant toute règle de réécriture.

- L'instruction **Options +FollowSymlinks** est facultative mais peut servir dans certaines configurations.

- L'instruction **RewriteEngine on** indique que nous souhaitons utiliser le module de réécriture d'URL. Si vous avez un problème avec une règle de réécriture que vous venez d'ajouter, vous pouvez désactiver en quelques secondes la réécriture d'URL le temps de comprendre le problème : il vous suffit d'écrire **RewriteEngine off** à la place de **RewriteEngine on**.

La suite du fichier est constituée d'une série de règles de réécriture. Chaque règle est écrite sur une seule ligne (sauf règles complexes) et respecte le format suivant :

**RewriteRule** URL\_A\_REECRIRE URL\_REECRITE

- **RewriteRule** est un mot-clé spécifique au module mod\_rewrite qui indique que la ligne définit une règle de réécriture.
- Ensuite vient l'URL à réécrire, c'est-à-dire l'URL « propre » sans existence physique sur le serveur.
- Enfin vient l'URL réécrite, c'est-à-dire l'URL telle qu'elle sera appelée en interne sur le serveur.

Le format de l'URL à réécrire est basé sur les expressions régulières, dont la base devra être acquise pour pouvoir définir des règles de réécriture. Ne vous inquiétez pas, pour la plupart des cas c'est très simple.

Voici la liste des éléments pris en considération dans les règles de réécriture :

Élément	Explications
^	Indique le début de l'URL à réécrire. Ce caractère est facultatif mais il est plus rigoureux de l'utiliser.
<b>article-</b>	Cette partie de l'URL n'est pas utilisée directement. Nous aurions pu écrire <b>art</b> à la place de <b>article</b> . Ce préfixe peut servir à différencier différents schémas d'URL, et il permet à l'internaute de mieux comprendre l'objet de la page.
( )	Les parenthèses servent à encadrer une variable dont la valeur est récupérée dans la 3ème partie de la ligne.
<b>[0-9]+</b>	Indique que la variable est composée d'un ou plusieurs chiffres.
\$	Indique la fin de l'URL à réécrire. Ce caractère est facultatif mais il est plus rigoureux de l'utiliser.
<b>/articles/</b>	Cette partie est parfois facultative (cela dépend de la configuration du serveur). En général il suffit d'indiquer l'emplacement du fichier de manière relative au répertoire dans lequel est situé le fichier .htaccess (donc on peut se passer de cet élément). Sur certains hébergeurs mutualisés (OVH ou Sivit pour n'en citer que deux), vous devez indiquer le chemin complet vers le fichier, à partir de la racine du site, comme dans notre exemple.
<b>article.php</b>	Nom du fichier que le serveur doit utiliser pour afficher la page. C'est le nom d'un fichier qui existe physiquement et qui contient un script (PHP dans notre exemple) de gestion de la page dynamique.
?	Caractère obligatoire précédant la série de variables passées dans l'URL réécrite.
<b>id=\$1</b>	Indique que la variable nommée id prendra la valeur située dans la première paire de parenthèses.
&	Caractère utilisé pour séparer 2 variables dans l'URL réécrite.

<b>rubrique=\$2</b>	Indique que la variable nommée rubrique prendra la valeur située dans la deuxième paire de parenthèses.
<b>[L]</b>	Drapeau (option) signifiant « Last », indiquant au module de réécriture qu'il doit s'arrêter. Plus précisément, si l'URL de la page demandée par le visiteur correspond au schéma défini par cette règle, alors le module de réécriture d'URL ne doit pas examiner les autres règles situées dans le reste du fichier .htaccess. Il n'est pas toujours obligatoire mais il ne fera pas de mal !

Cet exemple de règle de réécriture permet déjà de gérer notre rubrique d'articles, mais il existe d'autres règles plus complexes.

**Nota** : le livre « Google, trucs de pros » (<http://www.livre-google.com/>) propose dans son chapitre « Le référencement de sites web dynamiques », la **syntaxe complète**, bien plus approfondie que dans cet article, à utiliser dans le fichier .htaccess. Vous trouverez également de nombreuses informations sur le site officiel d'Apache : [http://httpd.apache.org/docs/mod/mod\\_rewrite.html](http://httpd.apache.org/docs/mod/mod_rewrite.html)

#### 4. Modifier tous les liens internes

Maintenant que nous avons défini les schémas d'URL et créé les règles de réécriture, il reste à vérifier que dans tout le site, tous les liens utilisent le bon schéma d'URL.

En effet, les règles de réécriture du fichier .htaccess ne suffisent pas à ce que tout votre site soit au nouveau format, avec des URL propres ! C'est à vous de changer la façon d'écrire les liens, que ce soit dans des pages statiques ou dans des pages dynamiques.

Bien entendu, vous devez pouvoir sauter cette étape si vous incluez la gestion de la réécriture d'URL dès la création du site, puisque vous aurez pris soin de générer dès le début des liens aux bons formats.

#### 5. Mettre à jour et tester

Il est temps de tester ! Transférez tous les fichiers modifiés en ligne, y compris le fichier .htaccess, puis rendez-vous dans votre navigateur pour tester si la réécriture fonctionne.

Pour reprendre notre exemple, comparez ce que vous obtenez en allant sur :

**<http://www.notre-site.com/articles/article-12-2-5.html>**

et sur :

**<http://www.notre-site.com/articles/article.php?id=12&page=2&rubrique=5>**

Vous devriez avoir exactement la même page...

En cas de blocage complet du site (par exemple avec une erreur de type 500), n'oubliez pas qu'il suffit de supprimer le fichier .htaccess (ou d'annuler les dernières modifications) pour que tout revienne dans l'ordre.

Nous vous conseillons d'utiliser un logiciel de vérification des liens à l'intérieur de votre site (vous pouvez par exemple choisir Xenu's Link Sleuth, à télécharger sur <http://home.snafu.de/tilman/xenulink.html>). Ce type de logiciel agit comme Googlebot : il parcourt vos pages en suivant tous les liens qu'il trouve. S'il ne trouve aucun lien mort (un lien menant à une page introuvable), alors vous n'avez fait aucune erreur ni dans vos règles de réécriture ni dans vos liens internes. Sinon, corrigez en conséquence.

## **Optimisation automatique de toutes les pages**

Une fois que vous avez mis en place la réécriture d'URL sur tout votre site, il reste deux étapes pour terminer l'optimisation de votre site (du point de vue du référencement) :

1. Créer des liens vers toutes les pages.
2. Optimiser le code de **chaque** page dynamique.

### **1. Créer des liens vers toutes les pages**

Les sites dynamiques comportent bien souvent un grand nombre de pages. La mise en place de la réécriture d'URL permet une bonne indexation, mais ce n'est pas une condition suffisante pour que toutes les pages de votre site soient indexées. **En effet, il est nécessaire de créer les conditions pour que les robots des moteurs puissent accéder à vos pages en suivant les liens présents sur votre site.**

- Si vous avez une rubrique contenant des articles (actualité par exemple), prévoyez une zone d'archives avec des liens vers tous les articles, hiérarchisés de manière chronologique.
- Si vous avez un forum avec des milliers de discussions, vérifiez que tous les liens qui permettent de naviguer de page en page utilisent le bon format d'URL. Vous pouvez également prévoir là aussi une rubrique d'archives, avec des liens vers tous les forums et toutes les discussions des forums, le tout réparti sur autant de pages que nécessaire (limitez-vous à une centaine de liens par page environ, éventuellement un peu plus).
- Si vous avez un catalogue de produits, vous avez certainement classé ces produits en catégories, sur un ou plusieurs niveaux. Présentez ces produits sous la forme d'un annuaire qui permet de naviguer dans tout le catalogue avec des liens classiques `<a href>`. Cet annuaire peut être complété par un moteur de recherche interne, souvent très apprécié des internautes, et compatible bien entendu avec votre catalogue de produits.
- Une page « Plan du site » adaptée peut également être créée dans cette optique.

### **2. Optimiser le code de chaque page dynamique**

Une page dynamique n'est rien d'autre qu'une page HTML créée sur mesure par un script. En général une telle page repose sur un modèle de page, reprenant le design du reste du site, et comportant certaines zones dont le contenu est généré en effectuant des requêtes dans une base de données.

Nous avons vu dans les articles précédents de cette lettre comment optimiser le code d'une page HTML pour le référencement. Naturellement, vous pouvez faire la même chose avec des pages dynamiques.

Nous vous conseillons de prévoir sur toutes ces pages les zones suivantes, à remplir de manière unique pour chaque page :

- \* Le titre (balise `<title>`).
- \* Les renseignements sur la page, utiles par exemple aux annuaires et dans une moindre mesure pour les moteurs (balises `<meta description>` et `<meta keywords>`).
- \* Un titre dans le contenu textuel, en haut de page (balise `<h1>`).
- \* Un texte descriptif qui résume la page, à placer le plus haut possible sur la page (balise `<p>`).
- \* Des liens vers les pages connexes (balise `<a>`).

Si vous respectez ces consignes, vous disposerez rapidement d'un site dont les milliers de pages seront indexées et toutes optimisées pour le référencement !

## **Conclusion**

Les sites dynamiques sont aujourd'hui de plus en plus courant, car ils apportent une grande souplesse de gestion et permettent d'exploiter de grandes quantités d'information. Même si leur utilisation se démocratise, ils sont aujourd'hui encore bien souvent mal conçus en terme de référencement, puisqu'ils cumulent souvent de nombreux facteurs bloquants (identifiants de session, URL complexes).

La mise en place de la réécriture d'URL est un travail parfois long, complexe et technique, mais qui permet d'obtenir des résultats sans commune mesure avec les sites statiques. Une fois bien mise en place, la réécriture d'URL – associée à une optimisation dynamique des pages – permet bien souvent de positionner le site sur Google ou les moteurs de recherche « du marché » pour des milliers d'expressions plutôt que quelques dizaines comme c'est le cas habituellement avec les sites statiques.