Le SEO, c'est dans l'en-tête (HTTP) !

Par Aymeric Bouillat

Domaine :	Recherche	**Référencement**
Niveau :	Pour tous	**Avancé**

L'en-tête HTTP est l'espace qui contient des indications échangées entre le serveur et le navigateur, donc les robots des moteurs de recherche. La bonne utilisation de ces données dans une optique de référencement naturel peut vous faciliter la vie, en extrayant un certain nombre de directives du code HTML, ce qui peut s'avérer très intéressant pour faciliter une refonte, par exemple. Voici quelques cas dans lesquels ces en-têtes HTTP sont très utiles. Liste non exhaustive (et qui sera complétée dans de futurs articles)...

Depuis quelques années, Google accepte l'implémentation de plusieurs directives liées au référencement naturel dans les en-têtes HTTP. Ces directives peuvent concerner l'indexation de vos URL et le suivi (ou non) des liens qu'elles contiennent, des indications sur la similarité de certaines URL (DUST : *Duplicate URL, Same Text*) pour l'utilisation de la balise "canonical", ou encore permettre à Google de mieux identifier les différentes versions linguistiques et géographiques de vos URL *via* l'attribut hreflang.

L'utilisation des en-têtes http pour effectuer ces déclarations auprès de Google est utile pour les fichiers PDF, DOC ou encore JPG dont le type Mime ne permet l'insertion de balise HTML <meta> ou <link> dans le corps des documents. Ces en-têtes sont également utiles dans d'autres cas. En effet, leur implémentation pouvant être faite directement côté serveur (Apache ou encore Nginx, par exemple), permet d'éviter de faire des modifications dans les templates/fonctions du CMS, la manipulation de ces balises <meta> (robots) ou <link> (canonical et hreflang) pouvant parfois être assez délicate.

Nous détaillerons dans cet article les différentes en-têtes supportées par Google ainsi que leur implémentation côté serveur pour des raisons de rapidité d'intégration et de performances, bien que la mise en place de ces dernières soient également possible *via* l'applicatif (ASP, PHP, etc.). Les exemples de code concerneront le serveur Web Apache, qui reste le plus courant. Afin de manipuler ces en-têtes *via* le serveur Web, il vous faudra installer ou faire installer le module *mod_headers* ainsi que *mod_rewrite* d'Apache.

Rappel : fonctionnement d'un en-tête http

Quand votre navigateur appelle un fichier sur un serveur Web (html, image, css, js, etc.), il lui envoie entre autres des informations sur sa capacité à pouvoir gérer la réponse, et plusieurs informations additionnelles : encodage accepté, type de navigateur, langue du navigateur, referer, etc. En réponse, le serveur vous renvoie l'élément

correspondant à l'URL demandée, avec des informations supplémentaires dans les en-têtes comme la durée de vie du fichier dans le cache du navigateur, ou encore la date de dernière modification de la ressource, par exemple.

Ces « informations techniques », invisibles pour l'internaute mais capitales pour les navigateurs et crawlers, transitent dans les en-têtes HTTP, avant le corps de chaque fichier (son contenu).

Elles vont nous permettre de donner des directives à Google, sans modifier le corps de la réponse (à savoir le contenu HTML dans notre cas) ce qui s'avère extrêmement utile quand la configuration du site ne permet pas de modification du code applicatif de façon souple, ou que le type de contenu n'est pas de type HTML.

En-tête X-Robots-Tag

L'en-tête X-Robots-Tag est l'équivalent de la balise <meta name=robots> qui supporte ces principaux attributs :
• *index/noindex*= indexer ou non le contenu ;
• *follow* ou *nofollow* : suivre ou non les liens contenus dans la ressource ;
• *nosnippet* : ne pas afficher de snippet (meta description) dans les pages de résultats ;
• *noarchive* : ne pas permettre à l'utilisateur d'afficher la version en cache du contenu (commande « cache: » de Google).

Syntaxe de l'en-tête :
X-Robots-Tag : attribut1, attribut2

Nous allons nous intéresser à deux attributs qui sont extrêmement utiles pour ne pas indexer ou accélérer la désindexation de contenus (noindex), avant d'en restreindre le crawl dans le fichier robots.txt, ou limiter le crawl d'URL non pertinentes (nofollow). A noter que Bing supporte également l'en-tête X-Robots-Tag.

Voici plusieurs cas pour lesquels seulement quelques lignes de configuration d'Apache vous permettront de corriger des problèmes de duplication de contenu, sans aucune intervention sur la partie applicative de votre site. Cela pourrait permettre une implémentation plus rapide par votre DSI ou votre prestataire technique.

Duplicate HTTP vs HTTPS

Dans le cas où votre site est indexé sur le protocole HTTP et HTTPS, il est préférable de n'en faire indexer qu'une seule version. Vous pourriez également utiliser une balise "canonical", mais dans le cas d'un site ayant un volume important de pages, cela peut créer une problématique de performance dans un souci d'optimisation du crawl.

Désindexer /ne pas indexer de la version HTTPS (Apache) :
Si le protocole utilisé est HTTPS (port 443), alors on déclare un environnement que l'on nomme « headernoindex » pour toutes les URL HTTPS. On ajoute ensuite un en-tête http *X-Robots-Tag : noindex* qui sera envoyé pour les réponses du serveur répondant à cette condition.

```
# empêcher l'indexation d'un site en HTTPS
RewriteCond %{SERVER_PORT} 443
RewriteRule . - [E=headernoindex]
```

```
Header set X-Robots-Tag "noindex"
env=headernoindex
```

Désindexer/ ne pas indexer la version HTTP (Apache) :
A l'inverse, si l'on souhaite indexer uniquement la version HTTPS d'un site, il faudra bloquer l'indexation de la version http :

```
# empêcher l'indexation d'un site en
HTTP
RewriteCond %{SERVER_PORT} 80
RewriteRule . - [E=headernoindex]
Header set X-Robots-Tag "noindex"
env=headernoindex
```

Ne pas indexer/désindexer les URL contenant une chaîne de paramètres spécifique :

Si Google a indexé un grand nombre de pages avec des paramètres de tri (sortby, order, orderby & co) et que vous préférez les désindexer plutôt que d'utiliser une canonical (toujours dans un souci d'optimisation du crawl de Googlebot), voici le code à utiliser :

```
# empêcher l'indexation des URL
contenant un paramètre de tri ?order=
# empêcher le crawler de suivre les liens
contenus dans ces URL
RewriteCond %{QUERY_STRING}
order= [NC]
RewriteRule . -
[E=ordernoindexnofollow]
Header set X-Robots-Tag "noindex,
nofollow" env=ordernoindexnofollow
```

Si l'URL demandée contient le paramètre "order=", on déclare une variable d'environnement Apache pour renvoyer l'entête http correspondant, définie avec cette condition.

Désindexer/ne pas indexer des répertoires ou certains type de fichiers

Les sections <Files> et <Directory> d'Apache vous permettent de cibler des répertoires ou des extensions de fichier spécifiques afin d'en restreindre l'indexation. Cela peut être utile en cas d'indexation d'un répertoire privé ou d'un type de fichier reconnu par Gooogle mais non-HTML que vous voudriez ne pas faire indexer (la directive X-Robots-Tag dans ce dernier cas étant la seule alternative pour ce type de fichiers.)

Ne pas indexer certaines extensions :

```
#Bloquer l'indexation des fichiers Word
et PDF
<Files ~ "\.(pdfldocldocx)$">
Header set X-Robots-Tag "noindex"
</Files>
```

Bloquer l'indexation des fichiers contenus dans un répertoire :*

```
<Directory
/home/siteweb/images/habillagesite/>
Header set X-Robots-Tag "noindex"
</Files>
```

* Directive Directory à placer dans la configuration du serveur et non dans un .htaccess

Informations contradictoires : noindex vs index

Pour revenir aux fichiers HTML, il se peut que malgré l'ajout d'un en-tête X-Robots-Tag, une balise <meta robots> soit présente et générée via votre CMS. Dans le cas où 2 informations contradictoires seraient renvoyés entre

ces 2 méthodes d'implémentation (meta robots vs X-Robots-Tag), c'est le noindex qui l'emporte dans tous les cas, d'où l'intérêt d'utiliser l'en-tête X-Robots-Tag pour empêcher l'indexation de contenus *via* Apache, sans aucune intervention dans le code du site.

Il vous sera donc possible de cibler différentes URL pour gérer l'indexation via Apache, que ça soit pour certains formats d'URL *via* les RewriteRule, des répertoires entiers *via* la section <Directory>, ou encore des fichiers spécifiques en fonction de leur extension.

Vous pourrez également ajouter des valeurs au X-Robots-Tag si des directives sont déjà renvoyées par ailleurs, grâce au paramètre *append* (Header append X-Robots-Tag : nofollow), qui injectera la valeur dans l'en-tête correspondant.

En-tête Link avec attribut Canonical

Il est possible d'indiquer l'élément canonical dans les en-têtes HTTP.

Syntaxe :
Link:
<http://www.monsite.com/pdf/dossier-seo-et-entetes.pdf>; rel="canonical"

Pour un site dupliqué en HTTP et HTTPS par exemple : si certaines pages remontent bien en HTTP, que vous n'avez pas de problématique de crawl, et que vous souhaitez utiliser cette balise à tout prix plutôt que des 301, vous pourrez alors l'utiliser pour ne faire remonter que le site en HTTPS dans les pages de résultats :

```
#canonical HTTP pour les URL HTTPS
RewriteCond %{SERVER_PORT} !443
RewriteRule (.*) - [E=CANONICAL]
Header set Link
'<https://%{HTTP_HOST}e%{REQUEST_
URI}e>; rel="canonical"'
env=CANONICAL
```

La canonical via les en-têtes peut être manipulée comme X-Robots-Tag *via* l'applicatif (et le langage interprété côté serveur utilisé par votre site). Parmi les principaux usages : "canonical" pour les URL utilisant des paramètres de tri, ou encore URL comportant des paramètres de tracking qui ne modifient pas le contenu de la page.

Mais son intérêt *via* les en-têtes est surtout très adapté aux fichiers non-HTML comme nous l'indiquions précédemment. Voici 2 exemples d'utilisation de cette Canonical pour des types de fichiers spécifiques :

Fichiers PDF

Certains articles ou guides au format PDF (*Portable Document Format*) sont parfois disponibles en complément des pages HTML, et visibles dans les pages de résultats, comme la version PDF d'un dossier ou un tutorial téléchargeable, ou encore des versions imprimables de pages HTML.

Ces fichiers PDF peuvent rentrer en conflit de duplication de contenu avec leurs versions HTML, et c'est là que la Canonical rentrera en jeu : des fichiers PDF peuvent avoir de bonnes positions dans les pages de résultats, mais ne plus utiliser les versions HTML pose plusieurs problèmes (perte du suivi des utilisateurs, visiteur arrivant sur un fichier hors contexte, etc).

Nous allons donc continuer de faire indexer ces fichiers PDF pouvant avoir un bon positionnement, et injecter un en-tête Canonical dans les en-têtes vers les versions HTML (source : http://dejanseo.com.au/pdf-hack/).

Exemple pour un fichier PDF précis :

```
#on définit une canonical pour un fichier pdf vers sa version HTML
RewriteRule ^pdf/adobe-photoshop-CS6.pdf$ - [E=pdfphotoshop]
Header set Link '<http://www.monsite.com/tutoriaux/adobe-photoshop.html>; rel="canonical"' env=pdfphotoshop
```

C'est donc l'URL de la page HTML adobe-photoshop.html qui remontera dans les SERP de Google (la commande « info : » sur le fichier PDF vous le confirmera par la suite).

Fichiers images et CDN

Le contenu dupliqué n'existe pas que sur les pages HTML, les images peuvent également être victimes de ce type de filtre. L'utilisation de CDN (*Content Delivery Network*) pour améliorer les performances de vos sites peut devenir la source de contenu dupliqué pour vos images :
URL principale :
http://www.monsite.com/img/background.jpg
Même image via les CDN's :
http://cdn1.monsite.com/img/background.jpg
http://cdn2.monsite.com/img/background.jpg
http://cdn3.monsite.com/img/background.jpg

Le code source pouvant appeler ces différentes URL, elles risquent de se retrouver rapidement en doublon dans l'index de Google, quand le robot Googlebot-Image les indexera.

Pour indiquer à Google que ces images sont identiques, nous allons renvoyer un en-tête Canonical sur les images délivrées par le CDN vers la version WWW :

```
<Directory /home/www/img>
Header set Link '<http://www.monsite.com%{REQUEST_URI}e>; rel="canonical"'
</Directory>
```

En-tête Link avec attribut hreflang

Cet en-tête est l'équivalent de l'attribut hreflang dans l'élément <link>, qui permet de définir pour une URL donnée, ses différentes variantes relatives à différentes régions et langues pour un meilleur ciblage de son audience dans les résultats de Google.

Syntaxe :

```
Link: <http://de.monsite.com/>; rel="alternate"; hreflang="de"
```

Soit

```
Link: <URL>; rel="alternate"; hreflang="codepays-codelangue"
```

Les URL pouvant être différentes d'un ciblage géographique ou linguistique à l'autre, l'implémentation d'hreflang *via* les en-têtes http est moins pertinente côté serveur puisque vous devrez connaître pour chaque URL, ses équivalences dans d'autres langues. L'utilisation de l'applicatif pour cette méthode d'implémentation sera donc plus adaptée.

Via PHP :
< ?php header('Link: <http://de.monsite.com>; rel="alternate"; hreflang="de");?>

Dans le cas d'un site disponible dans plus de 2 pays (ou 2 langues), il faudra chaîner les informations dans le même en-tête Link, chaque information sur une URL devant être séparée par une virgule (,).

Exemple :
Link: <http://de.monsite.com/>; rel="alternate"; hreflang="es",<http://es.monsite.com/>; rel="alternate"; hreflang="es"

Une question d'intégration

Dans plusieurs cas de figure, l'utilisation des en-têtes http peut faciliter la gestion de certaines balises, que ce soit pour des balises meta robots ou canonical par exemple. Une implémentation complexe de ces informations dans le <head> de vos templates devrait pouvoir être facilitée par les en-têtes http : pour plus de souplesse et sans avoir à modifier un code qui se retrouverait effacé lors d'un upgrade de site. Gardez-le en-tête !

Aymeric Bouillat, *consultant SEO, Resoneo*
(http://twitter.com/aymerictwit ou http://www.yapasdequoi.com)