Katkıda Bulunanlar: Mehmet Durmaz

# Amaç

Sitenin arama motorlarındaki görüntülüğünü olumsuz etkilememesi açısından öğelerinden olan yönlendirme hatalarını gidermeyi amaçlar.

# Sitenin .htaccess dosyası nerde bulunur

Roots/Bedrock iskeletindeki sitelerde .htaccess dosyası {master/staging}/{current/shared}/www içinde bulunur.

# 301 Yönlendirme

## Durum Kodu (Moved Permanently) Nedir?

“301 Moved Permanently” durum kodu, genel olarak bir sayfadan diğer bir sayfaya yönlendirme olarak kullanılmaktadır. Örneğin, siz /verpacking/kutu-kutu-pense sayfasına giriş yapacak ziyaretçiler için bir yönlendirme aktif ederek bu ziyaretçileri /verpacking/kutu-kutu sayfasına yönlendirebilirsiniz. Bu şekilde kullanım sağlayarak dilediğiniz gibi ziyaretçilerinizi sayfalar arasında yönlendirebilirsiniz.

## 301 Durum Kodunun SEO’ya Etkisi Nedir?

Zaman zaman site üzerindeki bağlantı yapısını değiştirmek isteyebilirsiniz. Bu da değiştirmek istediğiniz sayfanızdaki sahip olduğunuz SEO otoritesini kaybetmeden yeni sayfanıza yönlendirmenize yardımcı olacaktır. SEO konusunda 301 yönlendirmesi oldukça faydalı ve sık sık kullanılan durum kodudur. Ayrıca linki değişen sayfalarda 404 hatasının önüne geçmiş oluruz ve kalıcı olarak yönlendirme sağlamış oluruz.

# HTACCESS İLE İLGİLİ BİLGİLENDİRME

Kaynak <<https://stackoverflow.com/questions/21195916/wordpress-multisite-htaccess-please-explain-the-functionality>>

## Yönlendirmenin htaccess içinde Kurulumu

|  |  |
| --- | --- |
| Varsayılan htaccess dosyası single site için aşağıdaki gibidir; | Varsayılan htaccess dosyası multi site için aşağıdaki gibidir; |
| #BEGIN WordPress  RewriteEngine On  RewriteRule .\* - [E=HTTP\_AUTHORIZATION:%{HTTP:Authorization}]  RewriteBase /  RewriteRule ^index\.php$ - [L]  RewriteCond %{REQUEST\_FILENAME} !-f  RewriteCond %{REQUEST\_FILENAME} !-d  RewriteRule . /index.php [L]  RewriteCond %{HTTP\_HOST} !^www\. [NC]  RewriteRule ^(.\*)$ https://www.%{HTTP\_HOST}/$1 [R=301,L]  #END WordPress | #BEGIN WordPress  RewriteEngine On  RewriteRule .\* - [E=HTTP\_AUTHORIZATION:%{HTTP:Authorization}]  RewriteBase /  RewriteRule ^index\.php$ - [L]  # add this line below : a trailing slash to /wp-admin  RewriteRule ^([\_0-9a-zA-Z-]+/)?wp-admin$ $1wp-admin/ [R=301,L]  # remove this line -> RewriteCond %{REQUEST\_FILENAME} !-f  # remove this line -> RewriteCond %{REQUEST\_FILENAME} !-d  # add those lines below-----------------------------------------  RewriteCond %{REQUEST\_FILENAME} -f [OR]  RewriteCond %{REQUEST\_FILENAME} -d  RewriteRule ^ - [L]  RewriteRule ^([\_0-9a-zA-Z-]+/)?(wp-(content|admin|includes).\*) wp/$2 [L]  RewriteRule ^([\_0-9a-zA-Z-]+/)?(.\*\.php)$ wp/$2 [L]  # add those lines above-----------------------------------------  RewriteRule . /index.php [L]  RewriteCond %{HTTP\_HOST} !^www\. [NC]  RewriteRule ^(.\*)$ https://www.%{HTTP\_HOST}/$1 [R=301,L]  #END WordPress |

## RewriteEngine On

Bu ifade ile yönlendirme  yazma motorunu (YYM) açar. Bu ifade başta olmazsa  .htaccess içinde işlenen kurallar gözardı edilir.

## RewriteBase /

Bu kural YYM'ye  hani göreceli yolun baz alınacağını söyle. Bu durumda hepsi oluyor.

RewriteRule ^index\.php$ - [L]

Bu kural şu anlama gelir: "İstek index.php ise bu  YYM'dan geçirin ve durun ([L]ast kuralı yazmayı durdurun."

RewriteRule^([\_0-9a-zA-Z-]/)?wp-admin$ $1wp-admin/[R=301,L]+

bu kural <[\_0-9a-zA-Z-] regex kuralına uygun falanca filanca> ve ardından "/" ve wp-admin ile eşleşirse adresin sonuna "/" ekler ve URL'ye yönlendirir. Örnek: <http://example.com/foo/bar/wp-admin>, <http://example.com/foo/bar/wp-admin/> adresine yönlendirilir

RewriteCond %{REQUEST\_FILENAME} -f [OR]

RewriteCond %{REQUEST\_FILENAME} -d

RewriteRule ^ - [L]

Bu kural, isteğin mevcut bir dosya veya dizin için olup olmadığını, YYM'dan geçirip yeniden yazmayı durdurduğunu söyler.

RewriteRule ^([\_0-9a-zA-Z-]/)?(wp-(content|admin|includes).) wp/$2 [L],\*+

Bu kural wp-content veya wp-admin veya wp-include ifadelerininden önce gelen ifadeleri çıkarır ve yeniden yazar Örnek: [http://example.com/blah/blah/wp-admin /some/more](http://example.com/blah/blah/wp-admin%20/some/more), /wp-admin/some/more olarak yeniden yazılır.

RewriteRule ^([\_0-9a-zA-Z-]/)?(.\.php)$ wp/$2 [L]\*+

Bu kural, <birşeyler> ardından gelen herhangi bir php dosyasıyla eşleşir ve URI'yi <birşeyleri>'i çıkarıp yeniden yazar. Örnek: <http://example.com/abc/123/somefile.php> /somefile.php dosyasına yeniden yazılır.

RewriteRule . index.php[L]

Diğer kuralların hiçbiri uygulanmadıysa, bu kural diğer her şeyi /index.php dosyasına yeniden yazar.

Son olarak; dolar-1 2 mevzusu için;

RewriteRule ^([\_0-9a-zA-Z-]+/)?(.\*\.php)$ $2 [L]

Bir kullanıcının merhaba-there.php'ye gittiğini istediğini varsayalım. Cevap sana.php olacak. İkame dizginizde $1, ilk parentez setinin (merhaba) içeriğini içerirken, $2 ikinci setin (sana) içeriğini içerir. Normal ifadenizde parantez yakalama kümeleri olduğu gibi, ikame dizginizde her zaman tam olarak "dolar" değeri olacaktır.

Ancak İç içe parenleriniz varsa, örneğin (([a-z])-[a-z]), $1 her zaman en dıştaki yakalamaya atıfta bulunur (bu durumda tüm regex), $2 ilk iç içe kümedir, vb.

## NGINX Web sunucu ayarları

server {  
listen 80 default\_server;  
listen 443 ssl default\_server;

root /var/www/html/www;

ssl\_certificate /etc/ssl/certs/master.crt;  
ssl\_certificate\_key /etc/ssl/certs/master.key;

include /etc/nginx/monitoring.conf;

index index.php index.htm index.html;

# Disable sendfile as per <https://docs.vagrantup.com/v2/synced-folders/virtualbox.html>  
sendfile off;  
error\_log /dev/stdout info;  
access\_log /var/log/nginx/access.log;

#REVISION POINT: #######################################################

#Attempt to rewrite wordpress in sub directory  
rewrite ^/wp/([\_0-9a-zA-Z-])/(xmlrpc\.php|wp-[0-9a-z-]\.php) /wp/$2;  
rewrite ^/wp/([\_0-9a-zA-Z-]+)/(wp-(admin|content|includes).\*) /wp/$2;

#Add trailing slash to \*/wp-admin requests.  
rewrite /wp-admin$ $scheme://$host$uri/ permanent;

#SEND EXPIRES HEADERS AND TURN OFF 404 LOGGING  
location ~\* ^.+.(xml|ogg|ogv|svg|svgz|eot|otf|woff|mp4|ttf|css|rss|atom|js|jpg|jpeg|gif|png|ico|zip|tgz|gz|rar|bz2|doc|xls|exe|ppt|tar|mid|midi|wav|bmp|rtf)$ { access\_log off; log\_not\_found off; expires max; }

#Pass uploaded files to wp-includes/ms-files.php.  
if ($uri !~ wp-content/plugins) { rewrite /files/(.+)$ /wp-includes/ms-files.php?file=$1 last; }

# Rewrite multisite '.../wp-.' and '.../.php'.  
if (!-e $request\_filename) { rewrite /wp-admin$ $scheme://$host$uri/ permanent; rewrite ^/[\_0-9a-zA-Z-]+(/wp-.\*) /wp$1 last; rewrite ^/[\_0-9a-zA-Z-]+(/.\*\.php)$ /wp$1 last; }

# REVISION POINT: #######################################################

# From wordpress demo global\_restrictions.conf

# Global restrictions configuration file.

# Designed to be included in any server {} block.  
location = /favicon.ico { log\_not\_found off; access\_log off; }

location = /robots.txt

{ allow all; log\_not\_found off; access\_log off; }

# Deny all attempts to access hidden files such as .htaccess, .htpasswd, .DS\_Store (Mac).

# Keep logging the requests to parse later (or to pass to firewall utilities such as fail2ban)  
location ~ /\. { deny all; }  
  
# Deny access to any files with a .php extension in the uploads directory  
# Works in sub-directory installs and also in multisite network  
# Keep logging the requests to parse later (or to pass to firewall utilities such as fail2ban)  
location ~\* /(?:uploads|files)/.\*\.php$ { deny all; }

location /

{ absolute\_redirect off; # This is cool because no php is touched for static content. # include the "?$args" part so non-default permalinks doesn't break when using query string try\_files $uri $uri/ /index.php?$args; }

# pass the PHP scripts to FastCGI server listening on socket  
location ~ \.php$ { try\_files $uri =404;

# REVISION POINT: Include the fastcgi\_params defaults provided by nginx include /etc/nginx/fastcgi\_params; fastcgi\_split\_path\_info ^(.+\.php)(/.+)$;

fastcgi\_pass unix:/run/php-fpm.sock; #REVISED AS BELOW: fastcgi\_buffers 16 16k;

fastcgi\_buffers 4 128k; # REVISED AS BELOW: fastcgi\_buffer\_size 32k;

fastcgi\_buffer\_size 128k;

fastcgi\_param SCRIPT\_FILENAME $document\_root$fastcgi\_script\_name; fastcgi\_param SCRIPT\_NAME $fastcgi\_script\_name;

fastcgi\_index index.php; include fastcgi\_params; fastcgi\_intercept\_errors off;

# fastcgi\_read\_timeout should match max\_execution\_time in php.ini # REVISED AS BELOW: fastcgi\_read\_timeout 10m; fastcgi\_read\_timeout 3600s; fastcgi\_param SERVER\_NAME $host; fastcgi\_param HTTPS $fcgi\_https;

}

location ~\* \.(png|jpg|jpeg|gif|ico)$

{ expires max; log\_not\_found off; }

location ~\* \.(js|css)$

{ expires -1; log\_not\_found off; }

include /etc/nginx/common.d/\*.conf;  
include /mnt/ddev\_config/nginx/\*.conf;  
}

# Yönlendirme örnekleri:

RewriteRule ^/shop/(.\*)$ /winkel/$1 [R=301,L] :

{root}/shop/{herhangi bir route} olarak gelen istekleri root}/ winkel /{herhangi bir route olarak yönlendir.

RewriteRule ^/wp-content/(.\*)$ /app/$1 [R=301,L]

{root}/wp-content/{herhangi bir route} olarak gelen istekleri root}/ app /{herhangi bir route olarak yönlendir.

RewriteRule /winkel/verpakking/karton-dozen/karton-vouwdoos-model-b-44x32x60cm-dubbelgolf-wit/

https://www.hotcoldshop.com/winkel/verpakking/karton-dozen/karton-dozen-2/vouwdozen-specifiek-bedrukt-voor-koel-en-diepvries-producten/ [R=301,L]

ilk kısımda 404’e düşen bu adresi asıl yeri olan ikinci kısma yönlendir.