Katkıda Bulunanlar: Mehmet Durmaz

# Amaç

AUTOGIT imkanı olmayan websitesini bedrock uyumlu hale getirmek, host alanına yüklemek, versiyon kontrol sağlamak, dağıtımını sağlamaktır.

# Siteyi bedrock/roots uyumlu yapmak

Uyumlu hale getirmek için sitedeki bütün tema ve eklentiler not alınır. Composer JSON üretici makrodaki REPOSITORY kısmında bu pluginlerin varlığı araştırılır. Yoksa eklenir. Eğer Premium plugin var ise ve kütüphanemizde yok ise TMM-DEV-T09 GITLAB Üzerinden Composer Paket Yönetimi talimatına uygun olarak paketler kütüphanemize yüklenir.

# Repo oluşturulması

Githubdaki scaffold reposundan yararlanılarak bir paket oluşturulur ve composer.json içine yüklenir. İskelet repo adresi: [git@github.com:The-Max-Media/tmm-scaffold.git](mailto:git@github.com:The-Max-Media/tmm-scaffold.git)

# Sitenin Kurulumu

Bu talimat bedrock'a çevirdiğiniz siteyi geçiş sitesinde asıl veri tabanının kopyası ile siteyi denediğinizi varsayar

Eğer mümkünse combellde yeni bir veritabanı açın. Yeni veri tabanının erişim bilgilerini bir kenara not edin. Eğer kota dolu ise mevcut veri tabanını yedekleyin. Örnekler themax.media sitesi üzerinden verilecek

* 1. SSH ile hedef host'a bağlan
  2. Ana dizine geçiniz;

$ cd

* 1. Eğer yeni veritabanı kullanılacaksa 9. Adıma geçin. Mevut veri tabanı kullanılacaksa mevcut sitenin dizinine geçelim;

* 1. Eğer site master site ise;

$ cd www

Eğer site yancı site ise

$ cd subsites/ themax.media

* 1. Ayar dosyasının içini görelim;

$ nano wp-config.php

Aşağıdaki bilgileri benzer bilgileri not edin;

define( 'DB\_NAME', 'ID221855\_themaxworld' );

define( 'DB\_USER', 'ID221855\_themaxworld' );

define( 'DB\_PASSWORD', 'CK7IE1Hd7Fu' );

$table\_prefix = 'wp\_a7355f2df5\_';

* 1. Anadizine geçiniz;

$ cd

Checkout dizini açın ve girin (mkdir: cannot create directory 'checkout': File exists hatası alırsanız zaten var demekti sıkıntı yok)

$mkdir checkout

$ cd checkout

* 1. Siteniz için dizin açın ve girin

$ mkdir themax.media

$ cd themax.media

* 1. Shared dizinini açın ve girin

$ mkdir shared

$ cd shared

* 1. .env dosyası oluşturun ve düzenleyin

$ nano .env

EKLE:

DB\_NAME=ID221855\_TheMaxMediaV2

DB\_USER=ID221855\_TheMaxMediaV2

DB\_PASSWORD=5h8082825j0bjWBen310

DB\_HOST=ID221855\_TheMaxMediaV2.db.webhosting.be

DB\_PREFIX='wpi7\_'

WP\_ENV='production'

WP\_HOME='https://www.themax.media'

WP\_SITEURL="${WP\_HOME}/wp"

WP\_DEBUG\_LOG=/path/to/debug.log

WP\_MEMORY\_LIMIT='128M'

MEDIA\_TRASH=true

* 1. <https://roots.io/salts.html> adresinden girin ve ürettiğini anahtarları dosyanın sonuna girin. Diğer bilgileri ise not aldığınız bilgilere göre güncelleyin…

* 1. .gitignore dosyanızı inceleyin aşağıdaki girdilerin mevcut olduğundan emin olunuz. Dikkat: girdilerin en sonunda slash ‘/’ bulundurmayınız. Çünkü symlinkleri de exclude etmeniz gerekiyor aksi takdirde sonuçları beklenmedik olabilir:

www/app/debug.log

www/app/wp-cache-config.php

www/app/advanced-cache.php

www/app/aiowps\_backups

www/app/blogs.dir

www/app/cache

www/app/ewww

www/app/languages

www/app/mailimages

www/app/plugins

www/app/themes

www/app/upgrade

www/app/uploads

www/app/updraft

www/app/wpo-cache

www/app/webp-express

www/app/wflogs

www/app/wp-rocket-config

www/wp-content

www/.htaccess

composer.lock

.env

.DS\_Store

.buildpath

.project

.settings

Yukarıdaki girdileri tamamladıktan sonra current dizini içinde konsoldan aşağıdaki komutu çalıştırınız.

find \* -type l -not -exec grep -q "^{}$" .gitignore \; -print >> .gitignore

ardından .gitignore dosyanızın son satırlarına gözatıız. Eğer exclude yapmayı atladığınız bir giridi varsa o son satırlarda göreceksiniz

* 1. Root dizini oluşturun ve girin

$ mkdir www

$ cd www

* 1. htaccess dosyası oluşturun ve düzenleyin

$ nano .htaccess

EKLE:

#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

* 1. App dizini oluştuırun ve girin

$ mkdir app

$ cd app

* 1. Mevcut sitedeki içeriği kopyalayın. Eğer site master site ise;

$ cp -r ~/www/wp-content .

Eğer site yancı site ise

$ cp -r ~/subsites/themax.media/wp-content .

* 1. Kopyaladığımız dizini saçalım ve gerekmeyenleri silelim:

$ cd wp-content

$ mv \* ../.

$ cd ..

$ rm -rf wo-content

$ rm -rf plugins

$ rm -rf themes

* 1. Current dizini açalım

$ cd ../../..

$ mkdir current

$ cd current

* 1. Şimdi repomuzu klonlayalım (nokta dahil)

$ git clone {repo adresi} .

* 1. .env için simlink bağlantısı yapın

$ ln -s ~/checkout/themax.media/shared/.env .env

* 1. htaccess için bağlantı atılır

$ mkdir www

$ cd www

$ ln -s ~/checkout/themax.media/shared/www/.htaccess .htaccess

* 1. App içerikleri bağlanır. Mantık şu; shared/www/app içindeki tüm dizinleri ve dosyaları current/www/app içinde simlink ile bağlamamız gerekiyor. Olası tüm bağlamalar altta verildi; (örneğe konu olan dizinler mevcut değilse oluşturmalısınız=

ln -s ../shared/.env;

cd www;

ln -s ../../shared/www/.htaccess;

ln -s ../../shared/www/wp-content;

cd app;

ln -s ../../../shared/www/app/advanced-cache.php;

ln -s ../../../shared/www/app/debug.log;

ln -s ../../../shared/www/app/wp-cache-config.php;

ln -s ../../../shared/www/app/aiowps\_backups;

ln -s ../../../shared/www/app/blogs.dir;

ln -s ../../../shared/www/app/cache;

ln -s ../../../shared/www/app/ewww;

ln -s ../../../shared/www/app/languages;

ln -s ../../../shared/www/app/mailimages;

ln -s ../../../shared/www/app/upgrade;

ln -s ../../../shared/www/app/uploads;

ln -s ../../../shared/www/app/updraft;

ln -s ../../../shared/www/app/wpo-cache;

ln -s ../../../shared/www/app/webp-express;

ln -s ../../../shared/www/app/wflogs;

ln -s ../../../shared/www/app/wp-rocket-config;

cd themes;

ln -s ../../../mythemes/betheme-child;

ln -s ../../../mythemes/twentytwentyone-child;

ln -s ../../../mythemes/vantage-child;

* 1. Sitenin paketleri yüklenir

$ cd ../..

$ composer install

* 1. Diller güncellenir (kopyala yapıştır)

wp language theme install --all fr\_FR &&

wp language theme install --all en\_US &&

wp language theme install --all de\_DE &&

wp language theme install --all nl\_NL &&

wp language theme install --all tr\_TR &&

wp language core install nl\_NL &&

wp language core install en\_US &&

wp language core install de\_DE &&

wp language core install fr\_FR &&

wp language core install tr\_TR &&

wp language plugin install --all fr\_FR &&

wp language plugin install --all en\_US &&

wp language plugin install --all de\_DE &&

wp language plugin install --all nl\_NL &&

wp language plugin install --all tr\_TR &&

wp language theme update --all &&

wp language core update &&

wp language plugin update --all

* 1. Veri tabanı güncellenir. Eğer mevcut veri tabanı kullanılıyorsa bu işlem geçici arıza oluşturacak

$ wp search-replace wp-content app --all-tables

* 1. Anadizine geçiniz;

$ cd

* 1. Son adım; Eğer site master site ise;

$ mv www www\_beforeroot

$ ln -s ~/checkout/themax.media/current/www www

Eğer site yancı site ise

$ cd subsites/

$ mv max.media max.media\_beforeroots

$ ln -s ~/checkout/themax.media/current/www themax.media