Katkıda Bulunanlar: Mehmet Durmaz

## Ön Gereklilikler

DDev tarafından oluşturulan docker containeri üzerinde çalışan Roots/Bedrock klonu Wordpress yüklemesi.

## Tanımlamalar

VCS: Version Controling System. Bunun için Git kullanacağız.

WP: Wordpress.

Roots/Bedrock Klonu: Github üzerinden edinebildiğimiz WP’in git ile daha uyumlu hale getirilmiş versiyonu. <https://roots.io/bedrock/>

Lokal: Bilgisayarımız üzerindeki DDev tarafından oluşturulan docker containeri üzerinde çalışan Roots/Bedrock klonu çalışan WP sitemiz

Remote: Hosting firması üzerinde “Live” çalışan Roots/Bedrock klonu çalışan WP sitemiz.

Remote Server Automation Tool: Remote’a git ile push yapmak suretiyle deploy yapmayı imkan sağlayan araç. Bunu için Combell/Autogit kullanılıyor olacak.

Live Site: Son kullanıcının gördüğü internet üzerinde görülen site.

WordPress Packagist: WP eklenti ve temalarını composer ile yüklemeyi sağlayan sistem. Eklenti dosyalarının VCS içine yük olmasını engeller. https://wpackagist.org/

## Bedrock’a yüklenen eklentilerin hata vermesi durumunda…

Eğer yüklediğimiz bir eklenti ya da tema hata veriyorsa bu o yüklenen öğenin hard-coding ile kodlandığı ile ilgilidir. WP’de önemli dizinler dinamik olarak kodlanmalıydı.

Bedrock ile ilgili tüm ayarlar anadizindeki .env dosyasında yer alır. Başka yerde (ör: wp-config.php enviroment.php) arama.

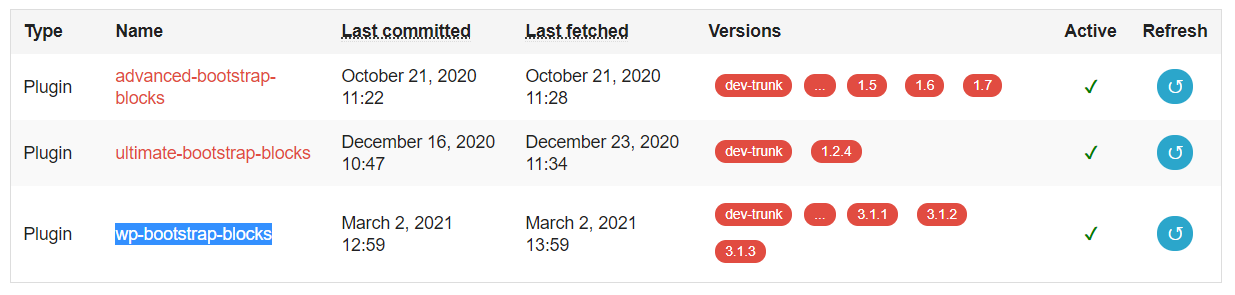
# Bedrock’a WP Packigst ile Plugin Yüklenmesi

Ana fikir şu: Eklentiler ve Zoraki (mu-plugin) eklentiler git tarafından ignore edilir. O yüzden composser.json dosyasında yerleştirmeleri gerekir ve bu yeterlidir. <https://www.smashingmagazine.com/2019/03/composer-wordpress/> adresinde verilen makalede güzel bir anlatım mevcut.

## Plugin yüklemek için:

Örneğin eski HCS sitemizdeki “Bootstrap Blocks By [Liip AG](https://liip.ch/)” yeni oluşumumuza VCS’ne dahil edebileceğimiz şekilde yüklemek istiyoruz.

1. <https://wpackagist.org/> sitesine girilir
2. Arama çubuğuna Bootstrap Blocks girilir ve aratılır.
3. Arama sonuçları: Name kolonundaki ibare paket ismimizdir.

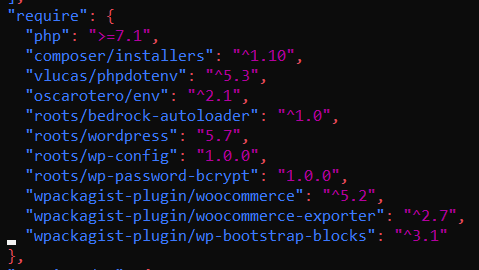


1. Şimdi Ubuntu konsolda WP klonumuzun olduğu klasöre girelim.
2. DDev’i başlatmadıysak başlatalım: ddev start
3. Konteynır içine girelim: ddev ssh
4. Gereken komut: composer require <namespace>/<packagename>

Burada namespace eklenti için: wpackagist-plugin , tema için wpackagist-theme olacaktır.

Bizim örneğimiz için: composer require wpackagist-plugin/wp-bootstrap-blocks

1. İndirme tamamlanacak ve gereklilik WP klonumuzun composer.json dosyasına işlenecektir:



1. Yüklediğimiz eklentiyi WP CLI ile aktif hale getirelim. Dikkat paket ismimiz burada da geçerli:

wp plugin activate wp-bootstrap-blocks yada wp plugin activate –all

theme için ise : wp theme activate

1. Bundan sonra yapmanız gereken VCS ile composer.json dosyanızı commit etmektir.
2. Bu haliye güncellediğiniz composer.json dosyanız artık diğer katılımcılar için de çalışacak. Onların tek yapması gereken ilk yükleme için composer install ya da Git güncellemesi ardından composer update komutlarını çalıştırmak.

## Pluginleri ve Sistemi Güncellemek için:

Bir eklenti composer.json dosyasına elle işlendiğinde, composer .json dosyasında değişiklik yapıldığımda ya da sistemi güncellemek için aşağıdaki komutunun çalıştırılması gerekir: composer update

## Normal bir eklentinin zoraki koşullanması

<https://roots.io/docs/bedrock/master/composer/#plugins> adresinde nasıl yapılacağı anlatılıyor.

Eğer paket wp packagist’e dahil değilse ama gerekiyorsa bunu composer ile yüklemenin yolu var: <https://github.com/composer/installers#custom-install-paths>