

Laravel 4

Türkçe Dokümantasyon

(Versiyon 4.1)



Hazırlayan ve E-kitap'a dönüştüren
Sinan Eldem

Laravel 4 Türke Dokümantasyon (v. 4.1) (Ücretsiz)

Laravel 4 Türkiye Forumları Çeviri Ekibi tarafından yapılan çeviriler

Sinan Eldem

Bu kitap şu adreste satılmaktadır <http://leanpub.com/laravel4-tr>

Bu versiyon şu tarihte yayımlandı 2014-08-04



This is a [Leanpub](#) book. Leanpub empowers authors and publishers with the Lean Publishing process. [Lean Publishing](#) is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

©2013 - 2014 Sinan Eldem

Kitabı tweetleyin!

Sinan Eldem'a kitabını şu adresten [Twitter](#) tanıtarak yardımcı olun!

Kitap için önerilen tweet:

Laravel 4 Türkçe Dokümantasyon kitaba dönüştürüldü. #laravel4tr @laraveltr @laravelphp

Kitap için önerilen hashtag [#laravel4-tr](#).

Kitap için diğerleri ne demiş merak ediyorsanız bağlantıya tıklayarak hashtagları arayabilirsiniz:

[https://twitter.com/search?q =#laravel4-tr](https://twitter.com/search?q=%23laravel4-tr)

İçindekiler

Artisan CLI	1
Giriş	1
Kullanım	1
Yardımcı (Helper) Fonksiyonları	3
Arrayler (Diziler)	3
Dosya Yolları	7
Yazı İşlemleri	8
URL İşlemleri	11
Diğer	12

Artisan CLI

Giriş

Artisan, Laravel içerisinde gelen CLI'ın (Command-line Interface) adıdır. Artisan size uygulamanızı geliştirirken birçok yardımcı komut sağlar. Artisan, güçlü Symfony Console bileşeni üzerinden geliştirilmiştir.

Kullanım

Tüm Kullanılabilir Komutların Listelenmesi

Tüm Artisan komutlarının bir listesini görmek için `list` komutunu kullanabilirsiniz:

```
1 php artisan list
```

Bir Komut için Yardım Ekranının Görüntülenmesi

Tüm komutların özel bir “yardım” ekranı vardır ve komut hakkındaki argüman sırası ile ayarlar gibi bilgilerin açıklanmasını sağlar. Bir yardım ekranını görüntülemek için komut adından önce `help` yazın:

```
1 php artisan help migrate
```

Yapılandırma Ortamının Belirtilmesi

`--env` anahtarını kullanarak bir komut çalıştırılırken kullanılacak olan yapılandırma ortamını belirtebilirsiniz:

```
1 php artisan migrate --env=local
```

Güncel Laravel Sürümünüzün Gösterilmesi

Ayrıca Laravel yüklemenizin güncel sürümünü de `--version` seçeneğini kullanarak görebilirsiniz:

```
1 php artisan --version
```

Yardımcı (Helper) Fonksiyonları

Arrayler (Diziler)

array_add

array_add fonksiyonu, verilen anahtar / değer çiftini, eğer daha önce eklenmemişse array'e eklemeye yarar.

```
1 $array = array('foo' => 'bar');  
2  
3 $array = array_add($array, 'key', 'value');
```

array_divide

array_divide fonksiyonu, birincisi anahtarlar, ikincisi değerler olacak şekilde iki farklı array döndürür.

```
1 $array = array('foo' => 'bar');  
2  
3 list($keys, $values) = array_divide($array);
```

array_dot

array_dot fonksiyonu, çok boyutlu bir array'i derinlikleri 'nokta (dot)' notasyonunu sağlayacak şekilde 1 boyutlu array'e çevirir.

```
1 $array = array('foo' => array('bar' => 'baz'));  
2  
3 $array = array_dot($array);  
4  
5 // array('foo.bar' => 'baz');
```

array_except

array_except fonksiyonu, verilen anahtar / değer çiftini array'den siler.

```
1 $array = array_except($array, array('keys', 'to', 'remove'));
```

array_fetch

`array_fetch` metodu seçilen bir iç elemanı içeren düz bir dizi döndürür.

```
1 $array = array(  
2     array('developer' => array('name' => 'Taylor')),  
3     array('developer' => array('name' => 'Dayle')),  
4 );  
5  
6 $array = array_fetch($array, 'developer.name');  
7  
8 // array('Taylor', 'Dayle');
```

array_first

`array_first` fonksiyonu, verilen doğruluk testine uyan ilk array elemanını döndürür.

```
1 $array = array(100, 200, 300);  
2  
3 $value = array_first($array, function($key, $value)  
4 {  
5     return $value >= 150;  
6 });
```

Ayrıca varsayılan bir değer, üçüncü eleman olarak verilebilir:

```
1 $value = array_first($array, $callback, $default);
```

array_last

`array_last` metodu verilen doğruluk testine uyan son array elemanını döndürür.

```
1 $array = array(350, 400, 500, 300, 200, 100);
2
3 $value = array_last($array, function($key, $value)
4 {
5     return $value > 350;
6 });
7
8 // 500
```

Ayrıca varsayılan bir değer, üçüncü eleman olarak geçilebilir:

```
1 $value = array_last($array, $callback, $default);
```

array_flatten

array_flatten metodu çok boyutlu bir diziyi tek düzey halinde düzleştirir.

```
1 $array = array('name' => 'Joe', 'languages' => array('PHP', 'Ruby'));
2
3 $array = array_flatten($array);
4
5 // array('Joe', 'PHP', 'Ruby');
```

array_forget

array_forget metodu “dot” notasyonu kullanarak, derin bir iç içe diziden belirli bir anahtar / değer çiftini kaldıracaktır.

```
1 $array = array('names' => array('joe' => array('programmer')));
2
3 array_forget($array, 'names.joe');
```

array_get

array_get metodu nokta notasyonu kullanarak derin bir iç içe diziden belirli bir değeri döndürür.

```
1 $array = array('names' => array('joe' => array('programmer')));
2
3 $value = array_get($array, 'names.joe');
```

Not: array_get gibi birşey ama onun yerine nesneler mi istiyorsunuz? object_get kullanın.

array_only

array_only fonksiyonu, array'den sadece verilen anahtar / değer çiftlerini döndürür.

```
1 $array = array('name' => 'Joe', 'age' => 27, 'votes' => 1);  
2  
3 $array = array_only($array, array('name', 'votes'));
```

array_pluck

array_pluck metodu verilen bir anahtar / değer çiftleri listesini diziden koparacaktır.

```
1 $array = array(array('name' => 'Taylor'), array('name' => 'Dayle'));  
2  
3 $array = array_pluck($array, 'name');  
4  
5 // array('Taylor', 'Dayle');
```

array_pull

array_pull metodu diziden belirli bir anahtar / değer çifti döndürecek, aynı zamanda bu çifti diziden çıkartacaktır.

```
1 $array = array('name' => 'Taylor', 'age' => 27);  
2  
3 $name = array_pull($array, 'name');
```

array_set

array_set metodu nokta notasyonu kullanarak, derin bir iç içe dizide bir değer ayarlar.

```
1 $array = array('names' => array('programmer' => 'Joe'));  
2  
3 array_set($array, 'names.editor', 'Taylor');
```

array_sort

array_sort metodu bir diziyi verilen bir Closure sonuçlarına göre sıralar.

```
1 $array = array(  
2     array('name' => 'Jill'),  
3     array('name' => 'Barry'),  
4 );  
5  
6 $array = array_values(array_sort($array, function($value)  
7 {  
8     return $value['name'];  
9 }));
```

array_where

Bir diziyi verilen Closure kullanarak filtreler.

```
1 $array = array(100, '200', 300, '400', 500);  
2  
3 $array = array_where($array, function($key, $value)  
4 {  
5     return is_string($value);  
6 });  
7  
8 // Array ( [1] => 200 [3] => 400 )
```

head

Dizideki ilk elemanı döndürür. PHP 5.3.x'deki metod zincirleme işine yarar.

```
1 $first = head($this->returnsArray('foo'));
```

last

Dizideki son elemanı döndürür. Metod zincirlemesinde işe yarar.

```
1 $last = last($this->returnsArray('foo'));
```

Dosya Yolları

app_path

app dizininin tam dosya yolunu getirir.

```
1 $path = app_path();
```

base_path

Uygulamanın ana dizininin tam dosya yolunu getirir.

public_path

public dizininin tam dosya yolunu getirir.

storage_path

app/storage dizininin tam dosya yolunu getirir.

Yazı İşlemleri

camel_case

Yazıyı camelCase olacak şekilde düzenler.

```
1 $camel = camsel_case('foo_bar');  
2  
3 // fooBar
```

class_basename

Verilen class'ın namespace'ler olmadan sadece adını döndürür.

```
1 $class = class_basename('Foo\Bar\Baz');  
2  
3 // Baz
```

e

Verilen yazıya UTF-8 desteğiyle htmlentities fonksiyonunu uygular.

```
1 $entities = e('<html>foo</html>');
```

ends_with

Bir stringin verilen değerle bitip bitmediğini tespit eder.

```
1 $value = ends_with('This is my name', 'name');
```

snake_case

Yazıyı snake_case olacak şekilde düzenler.

```
1 $snake = snake_case('fooBar');  
2  
3 // foo_bar
```

str_limit

Bir stringin karakter sayısını sınırlar.

```
1 str_limit($value, $limit = 100, $end = '...')
```

Örnek:

```
1 $value = str_limit('The PHP framework for web artisans.', 7);  
2  
3 // The PHP...
```

starts_with

Bir stringin verilen değerle başlayıp başlamadığını tespit eder.

```
1 $value = starts_with('This is my name', 'This');
```

str_contains

Verilen yazının içinde verilen değer olup olmadığına karar verir.

```
1 $value = str_contains('This is my name', 'my');
```

str_finish

Verilen yazının sonuna verilen değeri ekler. Verilen değerden oluşan ekstralrı yok eder.

```
1 $string = str_finish('this/string', '/');  
2  
3 // this/string/
```

str_is

Verilen yazıyla verilen değerin eşleşip eşleşmediğine karar verir. Yıldız işareti (*) genel arama karakteri olarak kullanılabilir.

```
1 $value = str_is('foo*', 'foobar');
```

str_plural

Verilen kelimeyi çoğul hale getirir (Sadece ingilizce için geçerli).

```
1 $plural = str_plural('car');
```

str_random

Verilen değer kadar uzunlukta rastgele karakterlerden oluşan bir yazı üretir.

```
1 $string = str_random(40);
```

str_singular

Verilen kelimeyi tekil hale getirir (Sadece ingilizce için geçerli).

```
1 $singular = str_singular('cars');
```

studly_case

Verilen yazıyı StudlyCase olacak şekilde düzenler.

```
1 $value = studly_case('foo_bar');  
2  
3 // FooBar
```

trans

Verilen dil satırını çevirir. Lang : :get fonksiyonunun kısayolu.

```
1 $value = trans('validation.required');
```

trans_choice

Verilen dil satırını çekimli çevirir. `Lang::choice` fonksiyonunun kısayolu.

```
1 $value = trans_choice('foo.bar', $count);
```

URL İşlemleri

action

Belirli bir denetçi eylemi için bir URL üretir.

```
1 $url = action('HomeController@getIndex', $params);
```

route

Verilen isimli rota için URL oluştur.

```
1 $url = route('routeName', $params);
```

asset

Bir varlık için bir URL üretir.

```
1 $url = asset('img/photo.jpg');
```

link_to

Verilen URL'e gerekli HTML linkini oluşturur.

```
1 echo link_to('foo/bar', $title, $attributes = array(), $secure = null);
```

link_to_asset

Verilen varlık için bir HTML bağlantısı üretir.

```
1 echo link_to_asset('foo/bar.zip', $title, $attributes = array(), $secure = null);
```

link_to_route

Verilen rota için gerekli HTML linkini oluşturur.

```
1 echo link_to_route('route.name', $title, $parameters = array(), $attributes = arr\
2 ay());
```

link_to_action

Verilen bir denetçi eylemi için bir HTML linki oluşturur.

```
1 echo link_to_action('HomeController@getIndex', $title, $parameters = array(), $at\
2 tributes = array());
```

secure_asset

Verilen eleman için gerekli HTML linkini HTTPS kullanarak oluşturur.

```
1 echo secure_asset('foo/bar.zip', $title, $attributes = array());
```

secure_url

Verilen URL'e gerekli HTML linkini HTTPS kullanarak oluşturur.

```
1 echo secure_url('foo/bar', $parameters = array());
```

url

Verilen bir dosya yolu için tam kalifiye bir URL üretir.

```
1 echo url('foo/bar', $parameters = array(), $secure = null);
```

Diğer

csrf_token

CSRF token'inin güncel değerini döndürür.

```
1 $token = csrf_token();
```

dd

Verilen veriyi ekrana basar ve uygulamayı durdurur.

```
1 dd($value);
```

value

Eğer verilen değer anonim bir fonksiyonsa, değer olarak anonim fonksiyonun döndürdüğü değeri döndürür. Eğer değilse direkt değeri döndürür.

```
1 $value = value(function() { return 'bar'; });
```

with

Verilen nesneyi döndürür. PHP 5.3.x kullanımında metod zincirleme işlemi için çok yararlı.

```
1 $value = with(new Foo)->doWork();
```