

Laravel - Ein Framework für Webartisten

Allgemeines über Laravel

Einleitung

Herzlich Willkommen zu meinem kleinem Tutorial zu einem neuem Framework mit dem Namen [Laravel](#). Das Tutorial ist für Einsteiger ebenso Interessant wie für Fortgeschrittene. Ich denke man kann immer wieder in paar neue Sachen dazu lernen, und sich ein Bild machen wie andere Programmierer so arbeiten. Grundsätzlich gibt es bei meinem Tutorial natürlich immer auch andere Wege die gleiche Lösung zubekommen. Falls ihr Fehler entdeckt, oder Anregungen habt wie diverse Passagen besser gemacht werden können, dann zögert nicht mir diese mitzuteilen. Ich freue mich immer über Feedback.

Was ist Laravel

Vielleicht sollte erst einmal erklärt werden was Laravel ist. Grundsätzlich handelt es sich bei Laravel um ein sehr schnelles und leicht erlernbares PHP Framework. Egal ob für PHP Neulinge oder Vollzeit Programmierer. Laravel lässt alle Herzen höher schlagen.

Aber um Laravel mit den Worten von Taylor Otwell zu erklären:

“

A Framework For Web Artisans

Laravel is a clean and classy framework for PHP web development. Freeing you from spaghetti code, Laravel helps you create wonderful applications using simple, expressive syntax. Development should be a creative experience that you enjoy, not something that is painful. Enjoy the fresh air.

”

Laravel ist zu dem ausgezeichnet Dokumentiert, und die Community wächst von Tag zu Tag.

Unterschied zu anderen Frameworks

Der Hauptunterschied zu anderen Frameworks ist, das Laravel viele Eigenschaften von sehr beliebten Frameworks wie:

- Codeigniter
- Symfony
- CakePHP

benutzt. Wobei hier der Hauptaugenmerk auf dem Framework Symfony liegt. Desweiteren gibt Laravel dem Programmierer viele Freiheiten wie er seine Webapplikation umsetzen kann. Hierzu folgt später aber mehr.

Hervorzuheben sei noch die unglaublich schnelle und überaus herrausragende Template Engine mit dem Namen Blade. Blade erlaubt es euch in kürzester Zeit Views und Layouts umzusetzen, und auf Framework und Applikations Funktionen zuzugreifen ohne auch nur ein einzigstes PHP Tag zuschreiben.

Der Einstieg in Laravel

Installation

Die Installation von Laravel gestaltet sich als relativ einfach. Ihr ladet euch das Paket von der [Laravel Homepage](#) und kopiert den gesamten Ordnerinhalt, auf euren Webserver oder eure Lokale Umgebung. Nachdem ihr diesen Schritt durchgeführt habt, sollte euer Ordner Inhalt so aussehen.



Wichtig für euch ist zu wissen, dass ihr euren Webserver **immer** auf den Public Ordner verlinkt.

“

Der Public Ordner ist der einzige Ordner der aus dem WWW erreichbar sein darf

”

Voraussetzungen

Grundsätzlich benötigt ihr je nachdem was ihr erstellen wollt folgende Sachen um anzufangen:

- Webserver mit
 - Apache oder Nginx
 - mindestens PHP 5.3
 - eine MySQL, PostgreSQL, SQLite oder SQL Server Datenbank

Projekt Struktur

Die Ordnerstruktur haben wir etwas weiter oben schon behandelt. Gehen wir nun etwas weiter auf die einzelnen Ordner ein und klären dessen Funktion:

- **/application**

Dieser Ordner ist das Herzstück eurer Applikation. Er enthält das Routing, die Controller, eure Models und eure Views. Ihr werdet die meiste Zeit in diesem Ordner verbringen

- **/bundles**

Bundles sind der modulare Part von Laravel. Der größte Vorteil an Bundles ist, die einfache Installation und die einfache Weitergabe eurer Arbeit an andere Benutzer von Laravel. Ohne das diese Code in ihrer Applikation ändern müssen.

Der **/application** Ordner ist ebenfalls ein Bundle. Das sogenannte Default Bundle. Alles was ihr im **/application** Ordner findet, könnt ihr ebenfalls in euren eigenen Bundles benutzen.

- **/laravel**

In diesem Ordner befinden sich alle Kern Dateien von Laravel. Dies ist auch der einzigste Ordner welche bei einem System Update ersetzt werden muss. Wenn ihr einmal wissen wollt, wie eine bestimmte Methode funktioniert, könnt ihr einfach die entsprechende Datei öffnen und euch die Kommentare anschauen. Meistens erhält man schon daraus die komplette funktionsweise ohne lange im Forum oder in der Dokumentation suchen zu müssen.

- **/public**

Dies ist der Ordner welcher auch durch euren Webserver genutzt werden sollte. Er beinhaltet die bootstrap Datei *index.php* welche das Laravel Framework und den Routing Prozess startet. Ausserdem kann der Public Ordner dafür genutzt werden, CSS, Javascript Dateien, und Bilder bereitzustellen.

- **/storage**

Der Storage Ordner wird für alle Interaktionen genutzt, welche das Dateisystem als Treiber benutzen, z.B. Sessions und/oder Cache. Der Webserver muss Schreibrechte auf diesem Ordner haben.

In den meisten Fällen, musst du mit diesem Ordner überhaupt nicht interagieren, um eine Applikation mit Laravel zu erstellen

- **/artisan [file]**

Artisan ist Laravel's "Command Line Interface", welches dir ermöglicht zahlreiche Aufgaben über die Kommandozeile auszuführen. Natürlich auch deinen eigenen Aufgaben.

Um Artisan zu starten reicht ein einfaches

```
php artisan
```

auf der Kommandozeile in deinem Laravel Ordner.

- **/paths.php [file]**

Die `paths.php` wird vom Framework verwendet um die Pfade zu den obengenannten wichtigen Verzeichnissen zu erstellen. Du kannst dann ganz einfach mit

```
path('foldername')
```

den Pfad zum gewählten Verzeichnis nutzen. Es ist in den meisten Fällen nicht nötig, diese Datei zu bearbeiten.

Arbeiten mit Controllern

Grundsätzlich gibt es in Laravel 2 Möglichkeiten eine Webapplikation zu erstellen.

1. mit Controllern
2. mit Routen

Jedem ist freigestellt wie er seine Applikation aufbauen möchte. In diesem Abschnitt widmen wir uns aber den Controllern.

Routing Controllers

Zu allererst schauen wir uns mal einen einfachen Controller an:

```
class Account_Controller extends Base_Controller
{
    public function action_index()
    {
        echo "Das ist die Profil Seite";
    }

    public function action_login()
    {
        echo "Das ist das Login Formular";
    }

    public function action_logout()
    {
        echo "Das ist die Logout Action"
    }
}
```

Ein Controller ist eine PHP Klasse welche eine Sektion deiner Webseite oder deiner Web Applikation darstellt. Dessen Methoden oder 'Actions' repräsentieren eine individuelle Seite, oder einen Endpunkt von einem HTTP Request.

In unserem Beispiel repräsentiert der Account Controller die Benutzer Sektion unserer Webseite. Eine Profil Seite, eine Login Seite und eine Logout Seite.

“

Controller müssen immer auf Controller enden. Actionnamen müssen action vorangestellt bekommen. Desweitern muss der Controller immer vom Base_Controller, Controller oder

einen anderen Controller Klasse ableiten.

”

Unser Controller wurde als Datei im `/application/controllers` Ordner gespeichert. Der Name der Datei muss dem der Klasse in Kleinbuchstaben entsprechen:

```
/application/controllers/account.php
```

Doch bevor wir unseren Controller benutzen können, müssen wir diesen noch in `/application/routes.php` registrieren. Also fügen wir nun folgende Zeile ein:

```
Route::controller('account');
```

Sollte sich dein Controller in einem Unterordner befinden, dann kannst du diesen durch den Punkt Separator aufrufen.

```
Route::controller('in.einem.unterordner.account');
```

“

Die Tiefe der Verschachtelung ist hierbei absolut unbegrenzt.

”

Falls dein Controller in einem Bundle existiert, kannst du diesen ebenfalls aufrufen in dem du einfach den Bundlenamen gefolgt von 2 Doppelpunkten und dem Controllernamen aufrufst:

```
Route::controller('bundlename::account');
```

Nun kannst du dir das Ergebnis im Browser anschauen:

```
http://deine.domain/account/login
```

und schon kannst du sehen das die Ausgabe *Das ist das Login Formular* erscheint

“

Das ist nur erstmal ein Entwurf einer Übersetzung von daylerees.com Dabei handelt es sich um eine Sinngemäße Übersetzung von seinem Laravel Tutorial. Vielen Dank an dieser Stelle an Dayle Rees für seine Mühen.

”

