

Laravel

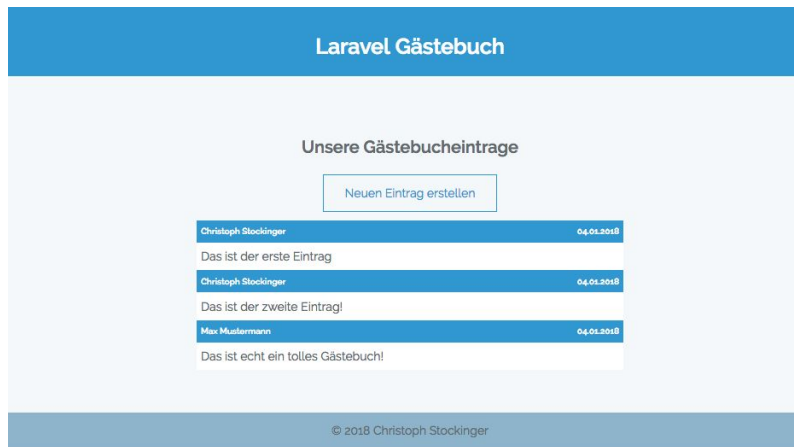
quelloffenes PHP-Webframework



Agenda

1. Allgemeines zu Laravel
2. Geschichte
3. Praxis - Entwicklung eines Gästebuchs

[https://github.com/
christophstockinger/laravelquestbook](https://github.com/christophstockinger/laravelquestbook)





Vorbereitung Praxis

1. Composer installieren

MacOS: `curl -sS https://getcomposer.org/installer | php`

2. Composer global zur Verfügung stellen

MacOS: `mv composer.phar /usr/local/bin/composer`

3. Prüfen der Installation

MacOS: `composer about`

Windows: `composer about`

Installation unter Windows über .exe-Datei

<http://getcomposer.org/Composer-Setup.exe>

Composer bereits vorhanden? Dann bitte aktualisieren:

`composer self-update`

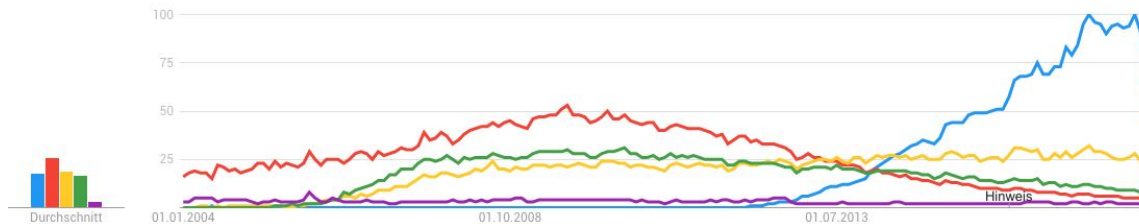
Allgemeines zu Laravel

- quelloffenes PHP-Webframework mit MVC-Muster (MIT-Lizenz)
- Initiator Taylor Ottwell im Jahr 2011
- Aktuelle Version 5.5.28 (28.12.2017)
- auf dem Weg das beliebteste PHP-Framework zu werden

Google Trends

2004 - heute

Laravel
Zend
Symfony
CakePHP
Yii





Geschichte

- Ottwell entwickelte Laravel als bessere Alternative zu CodeIgniter
- **Laravel 1:** - Unterstützung für Authentifizierung, Lokalisierung, Nutzersessions
 - Kein vollständiges MVC-Muster
- **Laravel 2:** - vollständig MVC-Konform
 - Unterstützung des Dependency Injection
- **Laravel 3:** - Kommandozeilentool namens Artisan
 - Unterstützung von verschiedenen Datenbanken
 - Eventverarbeitung
 - Bundles (Softwarepakete)
- **Laravel 4:** - Neuentwicklung auf Basis von Paketen (Verteilung über Composer)
- **Laravel 5:** - Scheduler für periodische Ausführung von Aufgaben
 - Abstraktionsschicht für den Zugriff auf Dateisysteme
 - verbesserte Handhabung von Paketressourcen
 - Integration externer Authentifizierungsverfahren

Praxis - Entwicklung eines Gästebuchs

- Laravel Projekt initialisieren

```
composer create-project laravel/laravel laravelguestbook --prefer-dist
```

- Projektordner betreten

```
cd laravelguestbook
```

- Projektordner in IDE öffnen

- Projekt starten

```
php artisan serve
```

→ Browser öffnen und angezeigter Pfad einfügen

Praxis - Entwicklung eines Gästebuchs

➤ Laravel Routing

`/routes/web.php`

Welche Seiten (Controller) brauchen wir?

- | | | |
|--|----------------------|--------------------------------------|
| - Startseite mit Ausgabe aller Einträge | → Startseite | = <code>IndexController@index</code> |
| - Seite zum Einträge hinzufügen mit Formular | → Eintrag hinzufügen | = <code>PostController@index</code> |

➤ Views

`/resources/views/`

Haupt-Template und Seiten-Views

- Ordner `layouts` erstellen → Datei `main.blade.php` anlegen
- Zwei weitere Ordner `index` und `post` erstellen → jeweils die Datei `index.blade.php` anlegen

(Ordner am besten so benennen wie Controller heißen und die Dateien wie die Events/Actions)

➤ globales Layout erstellen `main.blade.php`

CSS: `/resources/assets/sass/`

JS: `/resources/assets/js/`



Praxis - Entwicklung eines Gästebuchs

Generieren von Assets mit Laravel Mix

- NodeJS und npm installieren

NodeJS: <https://nodejs.org/de/download/>

npm: `npm install`

- Assets generieren

Laravel nutzt Sass → muss in CSS kompiliert werden

`npm run dev`

`npm run watch`

Damit werden die `app.css` und `app.js` erzeugt, welche im Layout eingebunden sind



Praxis - Entwicklung eines Gästebuchs

- Formular-Layout schreiben

`/post/index.blade.php`

- Controller und Methoden anlegen

`php artisan make:controller IndexController`

`php artisan make:controller PostController`

Die Controller findet man dann unter `/app/Http/Controllers/`

→ Anlegen einer öffentlichen Funktion namens `index` in beiden Controllern

- neue Route hinzufügen

`/routes/web.php`

- Post-Route für das abgesendete Formular

→ Eintrag hinzufügen

= `PostController@submit`



Praxis - Entwicklung eines Gästebuchs

- Absende-Layout schreiben
`/post/submit.blade.php`
- Methode in Controller hinzufügen
`/app/Http/Controllers/PostController`
→ Anlegen einer öffentlichen Funktion namens submit



Praxis - Entwicklung eines Gästebuchs

- Eloquent Model für Datenbank-Zugriff erzeugen

```
php artisan make:model Post -m
```

- Model wird erzeugt `/app/Post.php`
- Migrationsdatei (-m) wird auch erzeugt

```
/database/migrations/JAHR_MONAT_TAG_UHRZEIT_create_posts_table.php
```

- `up()`-Methode in der Migrations-Datei ergänzen

→ Hinzufügen der Formularfelder Name, E-Mail, Nachricht

```
$table->string('name');
```

```
$table->string('email');
```

```
$table->text('message');
```



Praxis - Entwicklung eines Gästebuchs

- XAMPP/ MAMP starten und neue Datenbank (PHPMyAdmin) anlegen
- Datenbank-Einstellungen konfigurieren
 - `.env` und `/config/database.php`
 - Ersetzen von `DB_DATABASE`, `DB_USERNAME` und `DB_PASSWORD`
 - ggf. Kollation der DB anpassen!
- Migration starten
 - `php artisan migrate`
- Einträge in Tabelle speichern
 - in der `submit()`-Methode ein neues Object Post erzeugen und die Request-Parameter an das Object übergeben.



Praxis - Entwicklung eines Gästebuchs

➤ Laravel Artisan Tinker

```
php artisan tinker
```

kleines Shell Tool in dem Laravel-Befehle ausgeführt werden können.

Einfach mal `App\Post::all();` eingeben

➤ Alle Einträge auf der Startseite ausgeben

```
/app/Http/Controllers/IndexController
```

1. Da wir ein eloquentes Model verwenden können wir uns ganz einfach mit folgendem Laravel-Befehl alle Datenbankeinträge in eine Datenbank speichern

```
$posts = Post::all();
```

Dabei sollte geachtet werden, dass die Variablen selbst erklären sind!

2. Alle Einträge an den View übergeben

Dazu erweitern wir die Return-Methode `view()`

```
return view ( 'index.index', [ 'posts' => $posts ] );
```

Praxis - Entwicklung eines Gästebuchs

➤ Blade-Template

`/resources/views/index/index.blade.php`

Auf ein Section verweisen:

`@yield('content')`

Verwendung von Blade-Templates in Child-Views:

`@extends('layouts.main')`

Eine Section definieren:

`@section('content') ... @endsection`

Blade-Foreach-Loop:

`@foreach ($posts as $post) ... @endforeach`

Blade-Statement :

`{{ $post->name }}`

Blade-PHP-Befehle:

`@php $date= new DateTime(...) @endphp`

Mehr zu Blade und weitere Befehle findet man unter

<https://laravel.com/docs/5.5/blade>

Und jetzt Lust auf Laravel bekommen?

Dokumentation: <https://laravel.com/docs/5.5>