

Datenbanken & Webtechnologien

Prof. Dr. Andreas Hannig

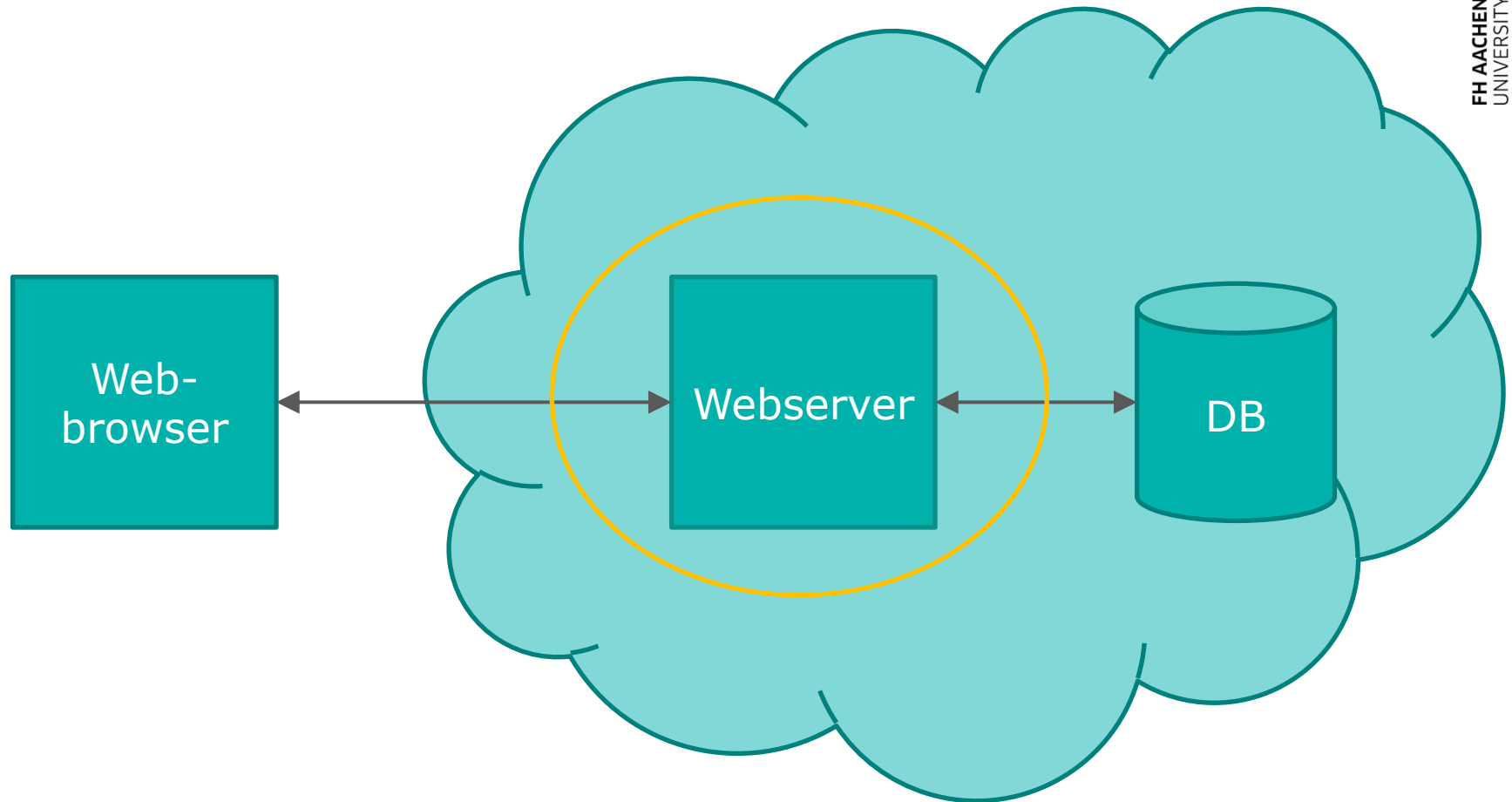
Web Application Frameworks



```
SELECT 'ho-ho-ho';  
echo 'ho-ho-ho';  
{{ 'ho-ho-ho' }}  
<em>ho-ho-ho</em>;
```

Datenbanken und Webtechnologien

Einordnung



Wir lernen den Einsatz von Web Application Frameworks anhand von Laravel kennen, womit sich komfortabel Web Anwendungen entwickeln lassen. Dazu wollen wir ...

- **Web Application Frameworks** einschätzen und **Auswahlkriterien** benennen können
- **Komponenten** von Laravel kennen und erklären können
- Ein **Routing** erstellen können
- Einen **Controller** erstellen können

Web Application Framework

Einführung

Ein Web Application Framework
ist ein **Software Framework**, das die
Entwicklung von
Webapplikationen
(inklusive Web Services, Web Resources, ...)
unterstützen soll.

- Dazu werden allgemeine, wiederkehrende Aufgaben von Anwendungen in Application Frameworks zusammengefasst.

Web Application Framework

Einführung

Ein **Web Application Framework** bietet in der Regel Lösungen unter anderem für die folgenden **wiederkehrenden Anforderungen** an:

Einheitliche Architekturvorgaben

URL Mapping / Routing

Caching

Web Templatesprache

Session Management

Authentifizierung

Datenbankzugriff

Scaffolding

Logging

Security

Queues

...

Web Application Framework

Einführung

- Es stehen **zahlreiche** Web Application Frameworks zur Verfügung. Einige Beispiel (für PHP) sind:



...



Anhand welcher **Kriterien** wählt man
ein Web Application Framework aus?

Web Application Framework

Blogs. Suchbegriff „Top OR best PHP Framework“

The Top 10 PHP Frameworks

From the advantages discussed above, it is evident that PHP frameworks will continue to be the server-side programming language of choice throughout 2019.

These frameworks, however, come in different sizes and shape. Your choice of PHP framework will depend on your level of experience, development timeframe, and level of experience among other factors.

The list below highlights the best PHP frameworks to expect in 2019. After reading through, I'm sure you will be comfortable enough to choose the best PHP framework 2019 that best suits your needs.



1. Laravel

This is one of the most popular open-source PHP frameworks, that was introduced in 2011. Laravel helps developers in building the most robust web applications by simplifying common tasks like caching, security, routing, and authentication.

The best PHP frameworks bellow are ordered by their popularity, technical specifications, and the ability to increase development efficiency:

1. Laravel

Love beautiful code? We do too.

The PHP Framework For Web Artisans



The top 10 PHP frameworks

Here's a list of the best PHP frameworks, based on popularity and ability to facilitate application development.

1. Laravel

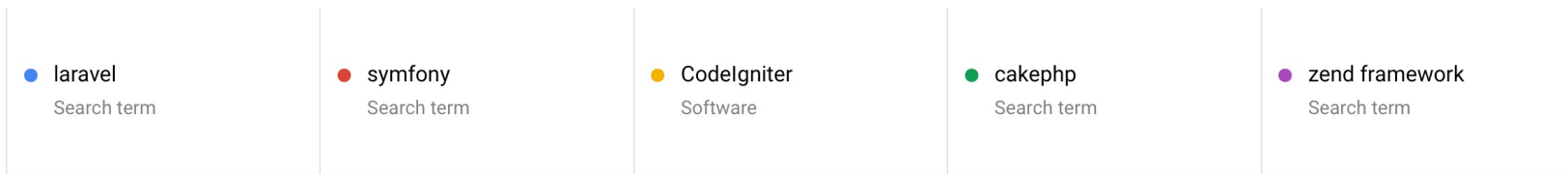


<https://www.dotcominfoway.com/blog/top-10-must-have-php-frameworks-in-2019/#gref>
<https://raygun.com/blog/top-php-frameworks/>
<https://www.hostinger.com/tutorials/best-php-framework>

16.08.2020

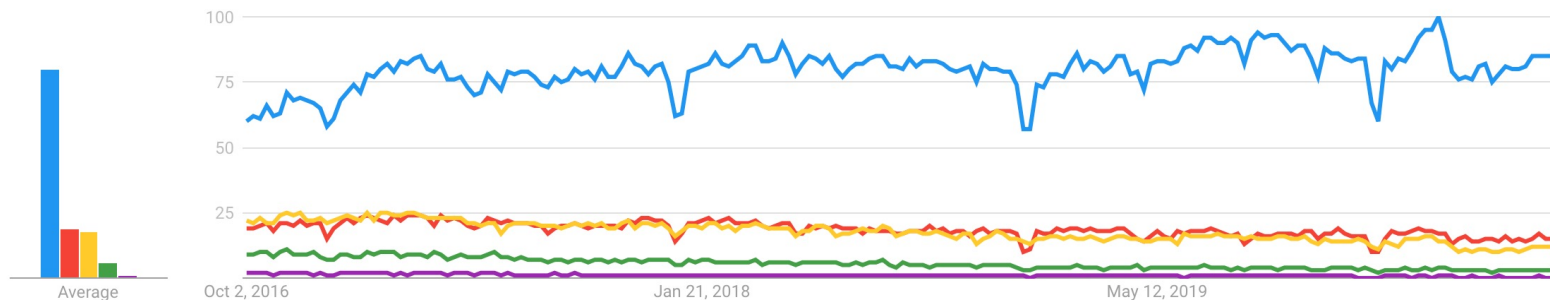
Web Application Framework

Google Trends



Worldwide ▼ 10/1/16 - 7/1/20 ▼ All categories ▼ Web Search ▼

Interest over time ⓘ



<https://trends.google.com/trends/explore?date=2016-10-01%202020-07-01&q=laravel,symfony,%2Fm%2F02qqdkj,cakephp,zend%20framework>

2.10.2020

Web Application Framework

Github - Stars

- Wie schätzt die **Entwicklungscommunity** das Framework ein?

Framework	Stars*
Laravel	56475
Symfony	22302
CodeIgniter	17780
Cakephp	8064
...	

Stand Dezember 2019

Stars*
62884
24330
18100
8211

Dez. 2020

Stars*
67569
26173
18200
8400

Dez. 2021

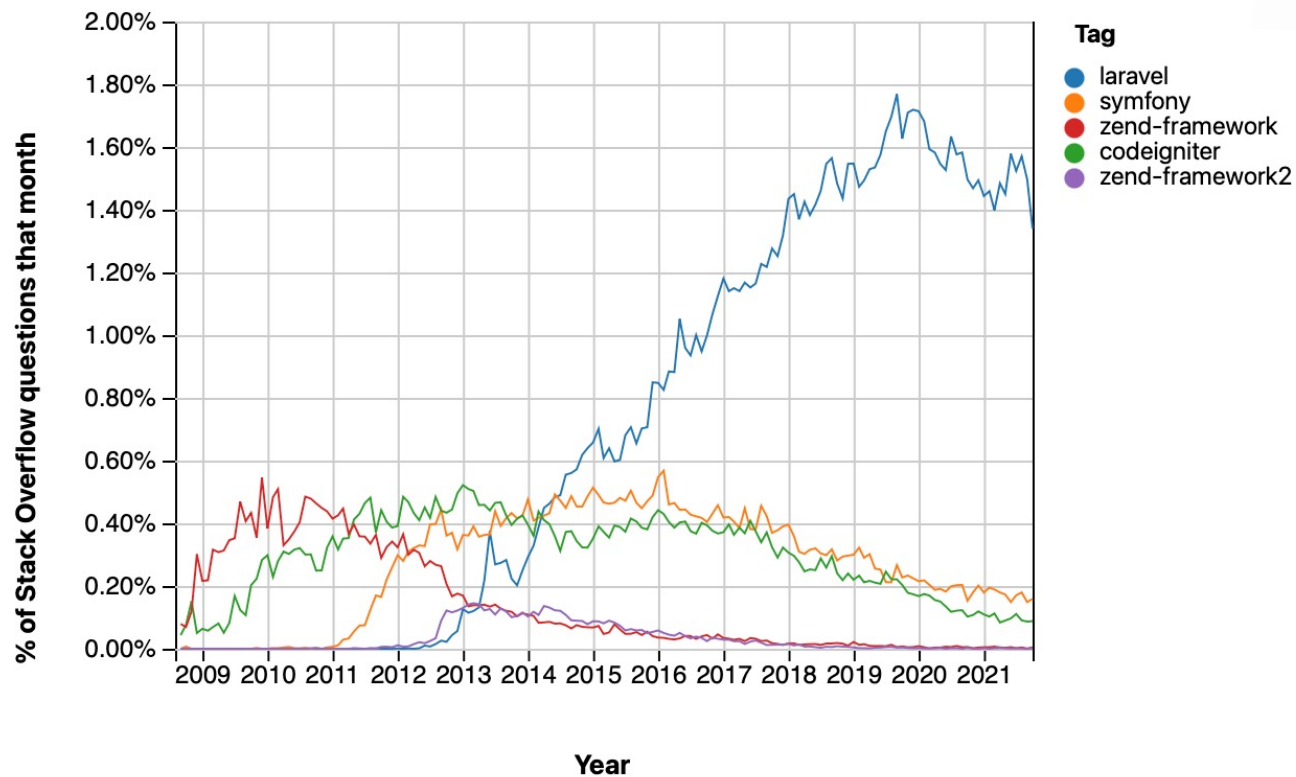
- Nutzer:innen können "Stars" verteilen zur Bekanntgabe, dass sie dieses Framework "mögen".

**) Nur die Stars des Top-Level-Projects*

<https://gitstar-ranking.com/laravel> 05.12.2021

Web Application Framework

Stackoverflow - Aktivität



<https://insights.stackoverflow.com/trends?tags=laravel%2Csymfony%2Ccodeigniter%2Czend-framework%2Czend-framework2>

8.11.2021

Web Application Framework

OpenHub - Aktivität



Laravel

Settings | Report Duplicate



Very High Activity

31

I Use This!

Analyzed about 1 hour ago, based on code collected about 4 hours ago.

Project Summary

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.

Tags

framework mvc webdev

Badges

build passing build passing

In a Nutshell, Laravel...

... has had 32,331 commits made by 2,876 contributors representing 128,130 lines of code

... is mostly written in PHP with an average number of source code comments

... has a well established, mature codebase maintained by a very large development team with stable Y-O-Y commits

... took an estimated 33 years of effort (COCOMO model) starting with its first commit in June, 2011 ending with its most recent commit about 21 hours ago

Quick Reference

Project Links: [Homepage](#) [Download](#)

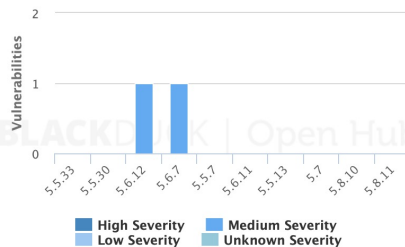
Code Locations: (2 Locations)

Similar Projects: [Joomla!](#) [Zend Framework](#) [FuelPHP](#) [Fat-Free Framework](#)

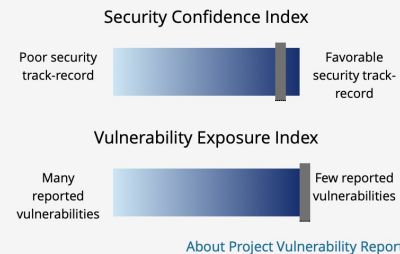
Managers: Become the first manager for Laravel

Project Security

Vulnerabilities per Version (last 10 releases)

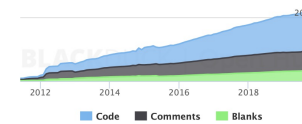


Project Vulnerability Report



Code

Lines of Code

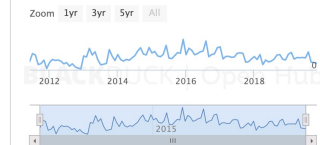


Languages



Activity

Commits per Month



30 Day Summary

Nov 6 2019 — Dec 6 2019
188 Commits
50 Contributors including 18 new contributors


12 Month Summary

Dec 6 2018 — Dec 6 2019
3486 Commits
Up + 212 (6%) from previous 12 months
487 Contributors
Down -84 (14%) from previous 12 months

<https://www.openhub.net/p/laravel> 6.12.2019

Web Application Framework

PHP Benchmarks

Release 5.8 9,490		PHP PHP 7.3 ▾			VERSION Laravel 5.8 ▾	
<h3>Hello world</h3> <div>4,890</div> <div>MEMORY: 1.7 MO</div> <div>Details</div> <div>Comparator</div>	1 CONCURRENCY	5 CONCURRENCY	10 CONCURRENCY	20 CONCURRENCY		
	 <div>1,399</div> <div>2 ms per request. 502 requests per second.</div>	<div>1,160</div> <div>2.1 ms per request. 2,353 requests per second.</div>	<div>1,201</div> <div>3.9 ms per request. 2,543 requests per second.</div>	<div>1,130</div> <div>7.8 ms per request. 2,557 requests per second.</div>		
<h3>REST API</h3> <div>4,600</div> <div>MEMORY: 4.3 MO</div> <div>Details</div> <div>Comparator</div>	1 CONCURRENCY	5 CONCURRENCY	10 CONCURRENCY	20 CONCURRENCY		
	<div>1,661</div> <div>5.5 ms per request. 181 requests per second.</div>	<div>1,053</div> <div>6.3 ms per request. 789 requests per second.</div>	<div>936</div> <div>12.3 ms per request. 810 requests per second.</div>	<div>950</div> <div>24.7 ms per request. 808 requests per second.</div>		



- Automatischer Test einer **vordefinierten Umgebung** (Minimalbeispiel „Hello World“) mit einer unterschiedlichen Anzahl **parallel ausgeführter Anfragen**.

<http://www.phpbenchmarks.com/en/> 6.12.2019

Web Application Framework

Zusammenfassung

- Bei **Open Source-Frameworks** zählen neben dem Funktionsumfang weitere **Auswahlkriterien**, wie:
 - Aktivität des Projekts
 - Qualität
 - Community
 - Verbreitung
 - Eingesetzte Technologien
 - Dokumentation
 - Verfügbarkeit professioneller Support



Web Application Framework

Diskussion

- Der Einsatz von Web Application Frameworks besitzen unter anderem die folgenden Vor- und Nachteile:

Vorteile

- Erhöhte **Entwicklungsgeschwindigkeit**
- Erhöhte **Qualität**
- Bessere **Skalierung und Performance**

Nachteile

- Notwendige **Einarbeitungszeit**
 - Höhere **Angreifbarkeit**
 - Erhöhte **Komplexität**

Web Application Framework

Hinweis zum Praktikum

- Im Praktikum Meilenstein 6 haben Sie die Wahl, ob Sie für die Lösung Laravel verwenden oder nicht.
- Laravel verwendet einige technische (wie Exceptions, Traits, ...) und fachliche (Services, Providers, ...) Themen mehr, als die Vorlesung umfasst.
 - Sie müssen sich selbst mehr aneignen.
- Sollten Sie das **Modul DBWT 2** wählen wollen, so empfehle ich Ihnen den Sprung zu Laravel, da dort Laravel eingesetzt wird und Sie die Erfahrung benötigen.



Laravel

... ein PHP Web Application Framework



Laravel

Einführung

- Laravel ist ein **MVC Web Application Framework**, entwickelt von Taylor Otwell. Erstes Release erschien im Juni 2011.
- Es bietet Implementierungen für alle der genannten **Features** von Web Application Frameworks.
- Verwendet MIT Lizenz (**Frei verwendbar** auch in kommerziellen Projekten)
- Bietet ein **umfangreiches Ökosystem** zur Entwicklung von Webanwendungen mit vielen weiteren Vorteilen.

*Wir verwenden **Version 8***

<https://github.com/laravel/framework> <https://laravel.com/> 14.09.2020

Laravel

Ökosystem (Auswahl)



Laravel Envoyer ist ein Deployment- (GitHub, Bitbucket) und **Monitoring-Tool** für Laravel-Server (**Zero-Downtime Deployment**)



Laravel Forge: **Provisionieren und Betreiben** von Laravel-Servern auf AWS, DigitalOcean und Linode.



Laravel Homestead

Laravel Valet und *Homestead* sind Entwicklungswerkzeuge zum **Testen** von Laravel-Applikationen.



Laravel Spark

Laravel Spark ist eine Erweiterung um **Benutzerverwaltung, Zahlungsmöglichkeiten, Teams, Zweifaktor Authentifizierung, ...**



Laravel Nova

Laravel Nova ist eine kostenpflichtige Erweiterung zur Erstellung von **CRUD Administrationspanels** (+Medienmanager, Menü-Erstellung, ...)

- Erweiterungen und Services stehen gegen Gebühr oder kostenlos zur Verfügung. Oft **amortisiert** sich der Einsatz bereits nach kurzer Zeit.



Prologue

Getting Started

Architecture Concepts

The Basics

Frontend

Security

Digging Deeper

Database

Eloquent ORM

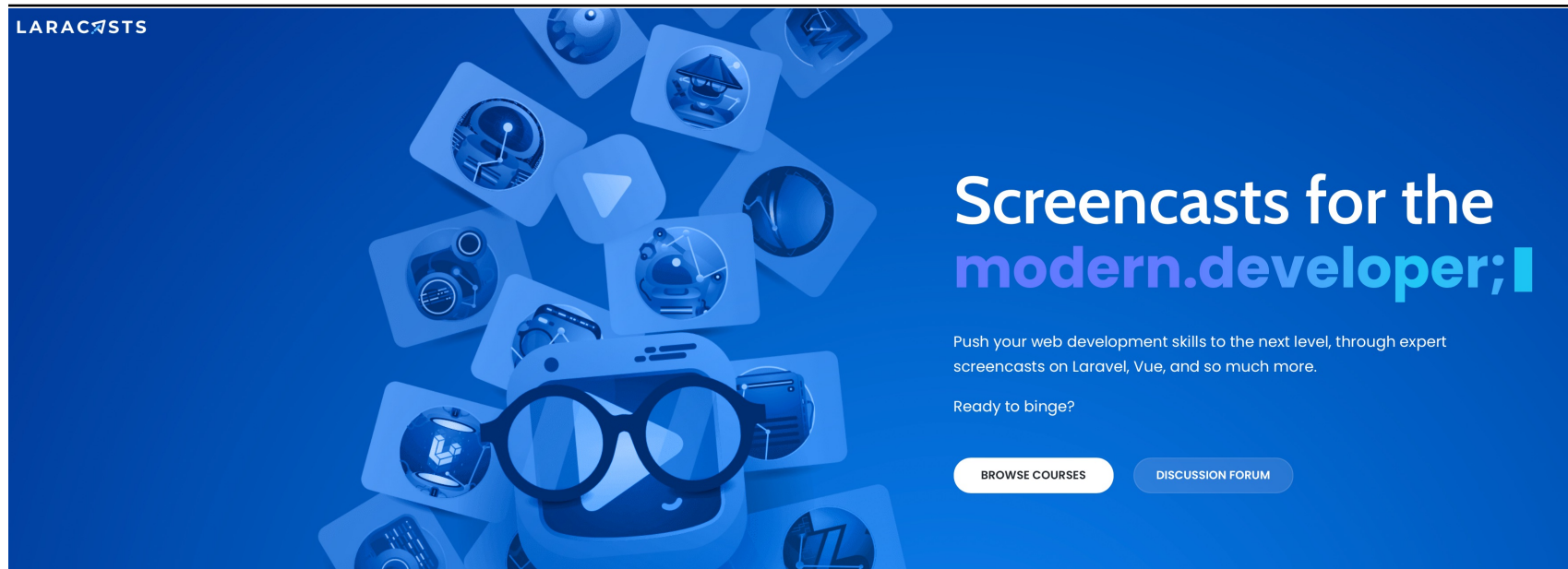
Testing

Official Packages

- Die Dokumentation von Laravel steht zur Verfügung unter:

<https://laravel.com/docs/8.x>

- Die zentralen Features werden gezeigt anhand von Anwendungsbeispielen.
- Weniger das „Big Picture“, sondern jeweils Einstiege in die einzelnen Komponenten
→ Mehr unter laracasts.com
- Den größeren Zusammenhang sowie die Verzahnung der einzelnen Bestandteile sehen wir in der Vorlesung.



- Viele hochwertige Lernvideos zu vielen Themen rund um Laravel.

Section 1 Prerequisites

- At a Glance
EPISODE 1 2:40 FREE
- 02 Install PHP, MySQL and Compo...
EPISODE 2 3:33 FREE
- 03 The Laravel Installer
EPISODE 3 3:02 FREE
- 04 Laravel Valet Setup
EPISODE 4 3:18 FREE

Frameworks

58 Series | 994 Videos

Languages

17 Series | 149 Videos

Techniques

36 Series | 402 Videos

Testing

12 Series | 100 Videos

Tooling

30 Series | 350 Videos

Route: get('/', 'PizzaCont
routes.php

<https://laracasts.com/skills/laravel> 10.12.2021

Laravel

Neues Projekt aufsetzen

Vorarbeit: Composer installieren

```
php path/to/composer.phar
```

```
create-project laravel/laravel projectname
```

*... erzeugt in dem Verzeichnis, in dem man gerade steht,
ein vollständig neues Laravel-Projekt mit dem Namen projectname.*

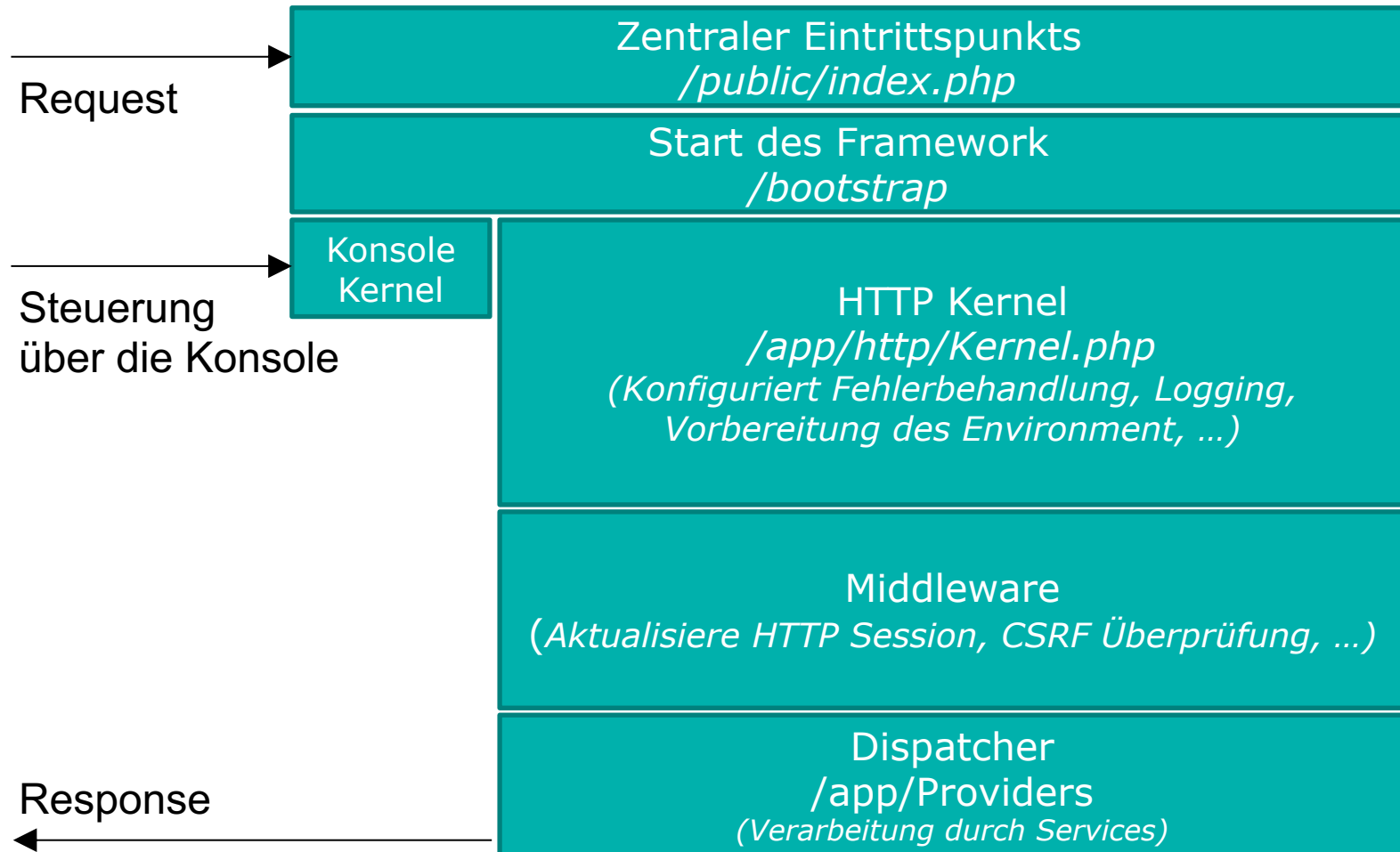
Laravel

Verzeichnisstruktur

Ordner/Dateien	Beschreibung
/app	Kernquelltexte der Anwendung (Controller, Models, ...)
/bootstrap	Startkonfiguration und Cache des Framework.
/config	Konfiguration der Anwendung
/database	Migrationen (Aktualisierung der Datenbankstruktur), Initialdaten, ...
/public	Startpunkt aller Anfragen an die Anwendung.
/resources	Views, Assets (wie LESS, SASS, JavaScript) und Sprachdateien
/routes	Routendefinitionen der Anwendung.
/storage	Geschriebene Daten: Cache. Übersetzte Blade-Templates, Datei-Caches, Datei-Sessions und Log-Nachrichten.
/tests	Testcodes für die Anwendung.
/vendor	Enthält die installierten Composer Pakete.
phpunit.xml	Konfiguration für Unittests.
.env	Umgebungskonfiguration
composer. {json,lock}	Composer Konfigurationsdateien
...	

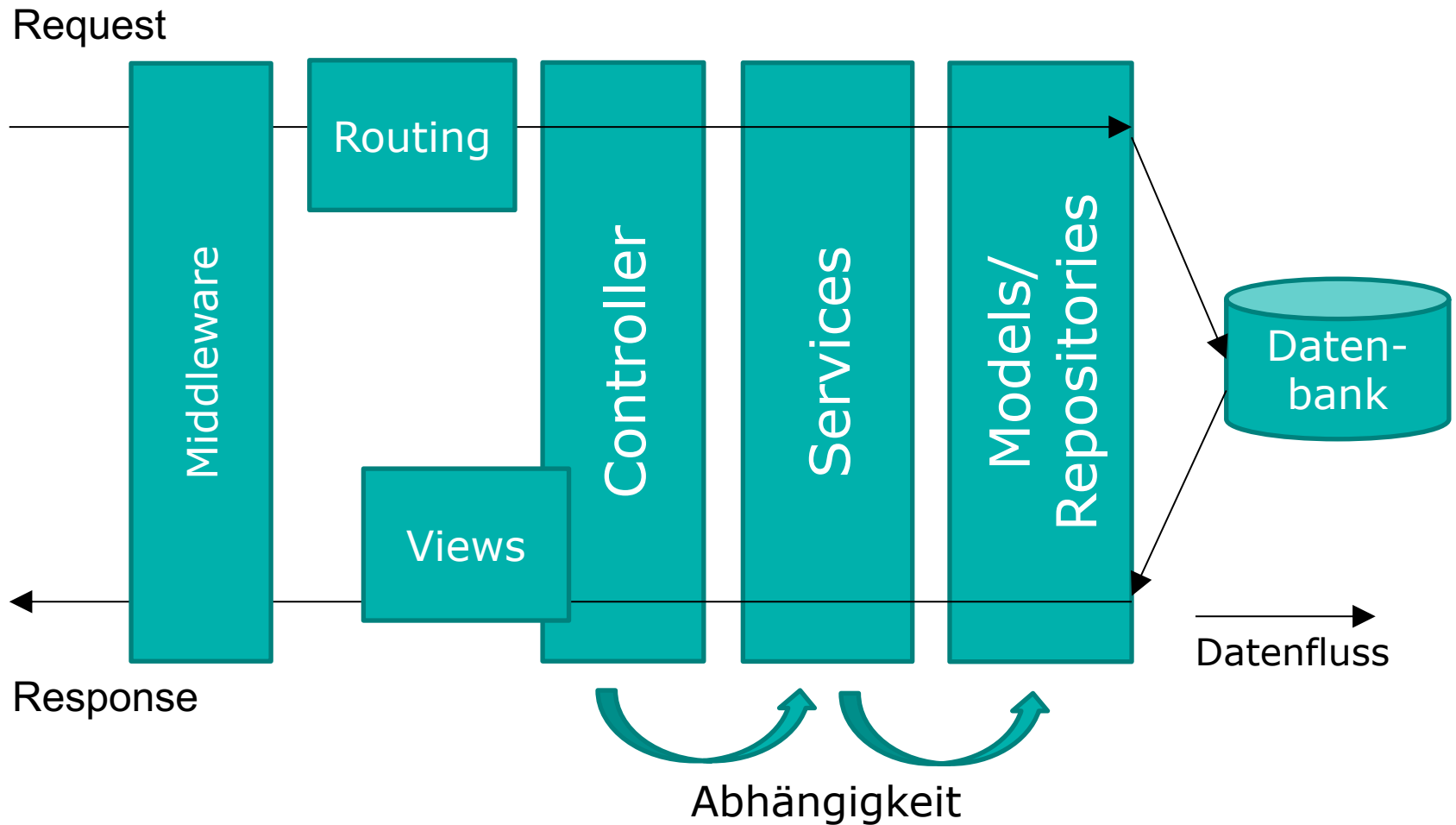
Laravel

Request Lifecycle



Laravel

Technische Ebenen



1. Konfiguration des Frameworks durchführen
2. Routen konfigurieren
3. Controller entwickeln
4. Middleware kennenlernen
5. Views anlegen
6. Datenbank konfigurieren
7. Datenbank in Controller verwenden

Nächste Vorlesung

Live-Demo

Live-Demo

Basiskonfiguration

Einstellungen über `.env` und `/config`

- Für die Konfiguration existieren zwei Orte:
das 1) /config-Verzeichnis und die 2) .env-Datei.
 - **/config enthält die Standardeinstellungen**, die durch Einstellungen in .env beim Start überschrieben werden.
(Produkteinstellungen, die initial für alle Installationen gelten sollen, werden hier konfiguriert)
 - **.env enthält die spezifische Konfiguration** der unterschiedlichen Umgebungen
(Gelten pro Entwicklungsumgebung: Jede/r Entwickler:in hat sein/e eigene .env lokal gespeichert.
Die Test- und Produktivumgebungen besitzt jeweils eine eigene .env)
→ Jede **Umgebung** besitzt eine **eigene** .env

Laravel / Basiskonfiguration

/config Inhalt

- Die Konfiguration von Laravel umfasst z.B. die folgenden Bestandteile:
 - app Anwendungseinstellungen
 - auth Authentifizierung
 - broadcasting Konfiguration von WebSockets
 - cache Cache
 - database Datenbankeinstellungen
 - filesystems Dateiablage
 - logging Log-Nachrichten
 - queue Asynchrone Jobausführung
 - session Sitzungseinstellungen
 - view Vieweinstellungen (Blade)
- ... und viele mehr ...

Laravel / Basiskonfiguration

.env Inhalt - Ausschnitt



- Die folgenden Parameter werden in der Datei .env konfiguriert:
 - **APP_NAME**
Name der Anwendung, der z.B. in Logfiles oder Fehlermeldungen dargestellt wird.
 - **APP_ENV**
Typ der Umgebung, z.B. „testing“, „development“, „local“ oder „production“.
 - **APP_KEY**
Zeichenkette (Schlüssel bzw. Salt) der z.B. für die Berechnung von Hashs verwendet wird.
 - **APP_DEBUG**= {true, false}
Einstellung, ob vollständige Fehlernachrichten (inkl. Stacktrace) an den Client verschickt werden sollen.
 - **APP_URL**
URL unter der die Anwendung erreichbar ist, wie „http://localhost“.
 - **DB_***
Verbindungskonfiguration der Datenbank.

Routing

Verbindung von externen Endpunkten mit interner
Verarbeitungslogik

Laravel / Routing

Einführung

- Das **Routing verbindet** die **Außenwelt** (Webbrowser, Serverprogramme, ...) mit der **Innenwelt** (Controller, Logik, ...)
- Das Routing verbindet konkrete externe Endpunkte (wie „/produkte“) mit internen (Controller-)Methoden.
(Haben wir bereits als `config/web.php` kennengelernt)

-

Das Routing umfasst die Bereiche:

- Webrouten (`routes/web.php`)
 - Webservices (`routes/api.php`)
 - WebSockets (`routes/channels.php`)
 - Console (`routes/console.php`)
- } Im Modul DBWT 2 enthalten

Laravel / Routing

Einführung

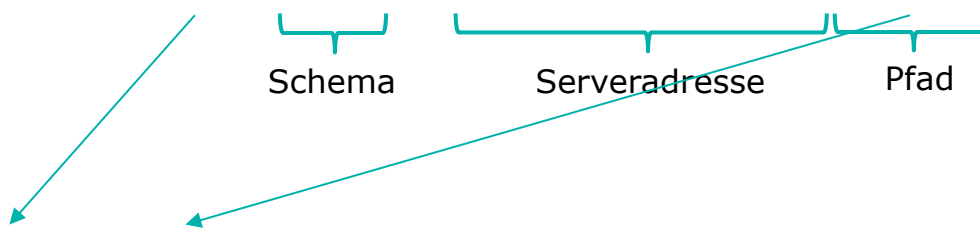
- Das Webrouting (routes/web.php) erlaubt die Registrierung von Endpunkten bzgl. der Webanwendung.
- Es gibt unterschiedliche **interne Routingziele**:
 - Anonyme Funktion
 - View
 - Redirect
 - Controller

Laravel / Routing

Beispiel mit anonymer Funktion

GET http://example.com/users

Schema Serveradresse Pfad



```
Route::get('users', function () {  
    return 'Hello World';  
});
```

Laravel / Routing

Dynamische Parameter

GET <http://example.com/users/216531?lang=de&filter=fa>

Schema

Serveradresse

Pfad

Abfrage / Query

```
Route::get('users/{id}', function ($id) {  
    return 'Hello User:' . $id;  
});
```

- Wenn der Parameter optional ist, kann dies mit einem nachgestellten „?“ gekennzeichnet werden:
`Route::get('users/{id?}', ...`
- Anstelle von `::get()` können auch andere HTTP Methoden verwendet werden, wie: `post`, `put`, `patch`, `delete`

Laravel / Routing

Views

- Routen, können direkt auf eine View verweisen, falls keine Vorverarbeitung durch einen Controller notwendig ist.

```
Route::view('welcome', 'welcome', [  
    'name' => 'Samantha']);
```

-- oder --

```
Route::get('welcome', function () {  
    return view('welcome', ['name' => 'Samantha']);  
});
```

Laravel / Routing

HTTP Verben & Redirect

- Routen mit mehreren HTTP Verben lassen sich auch komfortabler über `match()` setzen.

```
Route::match(['get', 'post'], '/path', function() {  
    return 'Hello World';  
});
```

- Alle HTTP Verben lassen sich über `any()` adressieren

```
Route::any('/path', function() {  
    return 'Hello World';  
});
```

- Weiterleitungen lassen sich mit einstellen z.B. mit:

```
Route::redirect('/here', '/to');  
Route::permanentRedirect('/here', '/to');
```

Laravel / Routing

Controller & Name

- Die Anbindung zu Methoden von Controllern geschieht z.B. über:

```
use App\Http\Controllers\ProdukteController;
```

```
Route::get('/produkt/{id}',  
           [ProdukteController::class, 'getProdukt']);
```

- Einer Route kann ein **Name** zugewiesen werden. Die Route kann dann später über diesen Namen (z.B. in Controllern oder Views bei Links) aufgelöst werden.

```
Route::get('user/profile', function () { /* ... */ })  
    ->name('profile');
```

Controller

Verarbeitung von Anfragen

Laravel / Controller

Einführung / Wiederholung

- Controller **verbinden externe Anfragen** mit internen **Verarbeitungslogik** und Datenmodellen. Sie übernehmen:
 - Steuern der Views (MVC)
 - Eingabevalidierung
 - Datentransformation
 - Delegation an Services
 - Steuern der Models
 - Sicherheitsprüfungen
- Controller werden in Laravel grundsätzlich als Klassen abgebildet. Diese befinden sich im Verzeichnis:

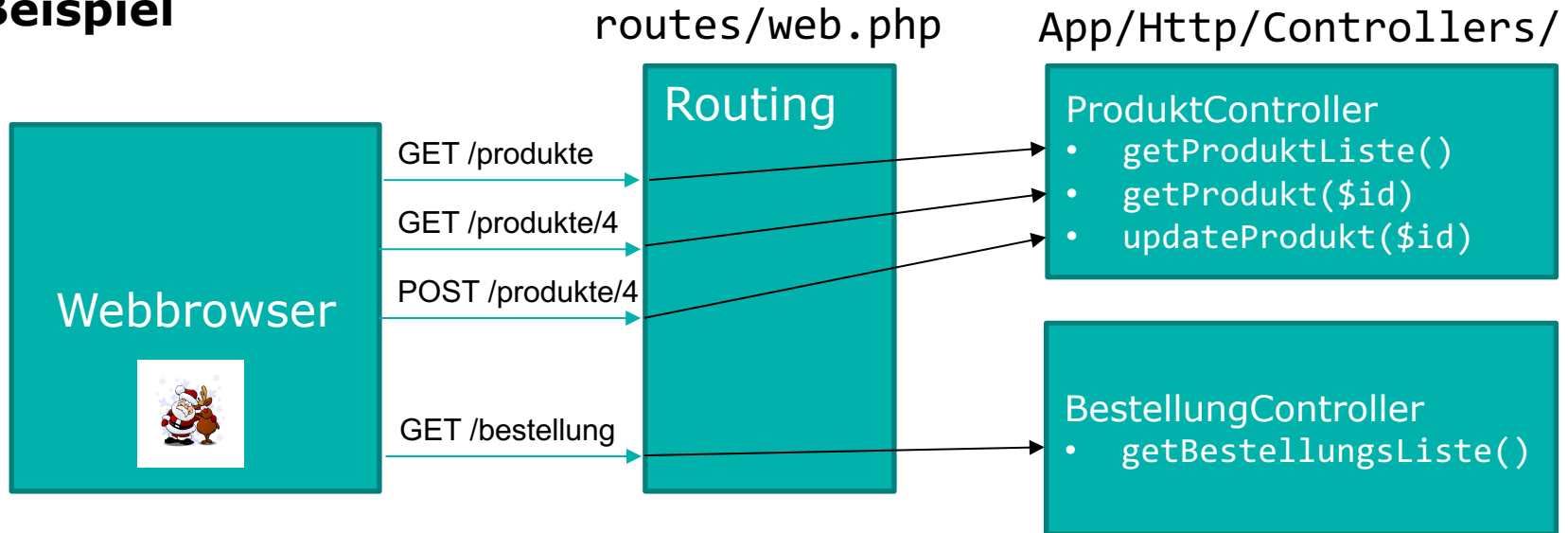
app/Http/Controllers

Laravel / Controller

Einführung / Wiederholung

- Controllerklassen bieten häufig mehrere **Actions** an. Eine Action ist eine Methode, die eine **Abfrage verarbeitet**.
- Controller beinhalten **logisch zusammengehörige** Actions.

Beispiel



- Die Methoden `getProduktListe()`, `updateProdukt($id)`, ... sind die Actions.

Laravel / Controller

Beispiel

```
<?php
```

```
/* Datei: app/Http/Controllers/ProdukteController.php */
```

```
namespace App\Http\Controllers;
```

```
class ProdukteController extends Controller {
```

```
    public function getProdukt($id) {
```

```
        return view('produkte');    → Führt das Blade-Template aus:  
    }                               /resources/views/produkte.blade.php
```

```
}
```

- Verbindung zwischen Endpunkt „/produkte“ und ProdukteController::getProduktListe() geschieht in der routes/web.php über:

```
use App\Http\Controllers\ProdukteController;
```

```
Route::get('/produkt/{id}', [ProdukteController::class, 'getProdukt']);
```

Web Application Framework & Laravel

Zusammenfassung

Wir haben Web Application Frameworks anhand des Beispiels Laravel kennengelernt, womit sich komfortabel Web Anwendungen entwickeln lassen. Dazu haben wir uns angeschaut:

- Web Application Framework
 - Allgemeine Informationen
 - Kriterien für die Auswahl
- Laravel Framework
- Ökosystem
- Komponenten
- Routing
- Controller

... lernen wir weitere **Bereiche von**
Laravel kennen womit wir
effizient **Webanwendungen** nach dem
MVC-Muster entwickeln
können.

... damit wir effizient wartbare und skalierbare Webanwendungen konstruieren können.



Das Team von DBWT wünscht Ihnen

Frohe Weihnachten

und einen

guten Rutsch

ins neue Jahr!



Kontrollfragen

Web Application Framework

- Welche Features können Web Application Frameworks anbieten? Was sind Vor- und Nachteile?
- Nach welchen Kriterien kann ein Web Application Framework ausgewählt werden?
- Nennen Sie 3 weitere Services aus dem Ökosystem von Laravel.
- Welche technischen Ebenen existieren im Laravel-Framework?
- Wie wird ein Routing konfiguriert? Was kann ein Routing adressieren?
- Was ist die Aufgabe von Controllern?

Datenbanken und Webtechnologien

FH Aachen

Fachbereich Elektrotechnik und Informationstechnik

Prof. Dr. Andreas Hannig

Lehrbereich Datenbanken und Business Intelligence

- <https://pixabay.com/de/photos/kakao-heiße-schokolade-zuckerstange-1908020/>