

Eine Webseite **MyWeb** soll verschiedenen Links auf anderen Webseiten bieten, Kontaktformulare an den Webmaster respektive Administrator senden und passwortgeschützt Daten unterhalten können. Im folgenden Text wird die Entwicklung von MyWeb Schritt für Schritt beschrieben.

Das **WWW**¹ ist ein über das **Internet**² abrufbares System von verknüpften autonomen **Websites**³ - autonomen Computer-Systemen. Es ermöglicht die Nutzung von **E-Mail**⁴, **FTP**⁵ und weiteren Diensten.

Eine Website, Webseite, Webanwendung wird im Internet über **HTTP**⁶ mit einer **URL**⁷ (Internet- oder Webadresse) lokalisiert.

Als Beispiel 'http://www.Informatik.FJKarli.ch'. Da Webadressen meist mit 'http://www' beginnen, kann dieser Präfix auch weggelassen werden. Einzelne Seiten einer Website (mit Texten, Bildern, grafischen Elementen, Videos, Tondokumente, usw.) werden üblicherweise mit **Hyperlinks**⁸ verknüpft.

Bei der Entwicklung einer Website oder Webanwendung profitiert man mit Vorteil von einem Grundgerüst – Framework. Das zur Zeit wohl umfangreichste ist wohl **Laravel**⁹.

Das Programm **Composer**¹⁰ wurde entwickelt, um von Laravel verwendeten Bibliotheken eines Projekts auf dem aktuellen Stand zu halten. Ohne Composer kann nicht sinnvoll mit Laravel gearbeitet werden. Die Installation unter Windows ist in der Datei Composer.pdf¹¹ beschrieben.

¹ **World Wide Web** (WWW)

² **Internet**: Die Verbreitung des Internets hat zu umfassenden Umwälzungen in vielen Lebensbereichen geführt. Es trug zu einem Modernisierungsschub in vielen Wirtschaftsbereichen sowie zur Entstehung neuer Wirtschaftszweige bei und hat zu einem grundlegenden Wandel des Kommunikationsverhaltens und der Mediennutzung im beruflichen und privaten Bereich geführt. Die kulturelle Bedeutung dieser Entwicklung wird manchmal mit der Erfindung des Buchdrucks gleichgesetzt.

Der Datenaustausch zwischen den verbundenen Rechnern erfolgt über die technisch normierten Internetprotokolle. Die Übertragung von Daten im Internet unabhängig von ihrem Inhalt, dem Absender und dem Empfänger wird als Netzneutralität bezeichnet.

³ Webseite, Homepage, Web-Anwendung

⁴ **E-Mail**, die „elektronische Post“ ist zum einen ein System zur computerbasierten Verwaltung von briefähnlichen Nachrichten und deren Übertragung über Computernetzwerke, insbesondere über das Internet. Zum anderen werden auch die auf diesem elektronischen Weg übertragenen Nachrichten selbst als E-Mails bezeichnet.

⁵ Das **File Transfer Protocol** (FTP,) ist ein 1985 spezifiziertes Netzwerkprotokoll zur Übertragung von Dateien über Internet.

⁶ Das **Hypertext Transfer Protocol** (HTTP) ist ein Netzwerkprotokoll zur Übertragung von Daten. Es wird hauptsächlich eingesetzt, um Webseiten (Hypertext-Dokumente) aus dem World Wide Web (WWW) in einen Webbrowser zu laden. Es ist jedoch nicht prinzipiell darauf beschränkt und auch als allgemeines Dateiübertragungsprotokoll sehr verbreitet.

⁷ Ein **URL** (Uniform Resource Locator) identifiziert eine Ressource, beispielsweise eine Website über die zu verwendende Zugriffsmethode (zum Beispiel das verwendete Netzwerkprotokoll wie HTTP oder FTP) und den Ort der Ressource in Computernetzwerken.

Im Internet werden URIs (Uniform Resource Identifier) zur Bezeichnung von Ressourcen (Dateien, Aufruf von Webservices, aber auch z. B. E-Mail-Empfängern) eingesetzt.

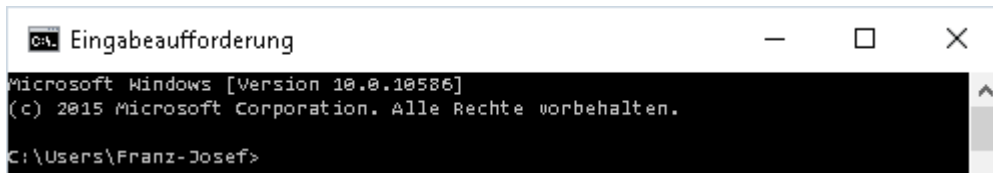
⁸ Mit einem **Hyperlink** respektive **Link** kann zu einem anderen elektronischen Dokument oder an eine andere Stelle innerhalb eines Dokuments gesprungen werden.

⁹ Das Framework **Laravel** ist eine Sammlung von Bausteinen, die optimal für die Entwicklung von dynamischen Webanwendungen ausgelegt ist. Durch vordefinierte und vorgefertigten Klassen werden sich wiederholende Tätigkeiten vereinfacht und die Wiederverwendung von Code und die Selbstdokumentation der Software-Entwicklung gefördert.

¹⁰ <https://getcomposer.org/> respektive <https://getcomposer.org/download/>

¹¹ [Http://Informatik.FJKarli.ch/PDF/Composer.pdf](http://Informatik.FJKarli.ch/PDF/Composer.pdf)

Bei der Entwicklung einer Webseite kommt man nicht um die Verwendung des CLI (Command-Line-Interpreter, Eingabeaufforderung oder DOS-Fensters) herum. Ohne weitere Vorkehrungen startet die Eingabeaufforderung im Benutzerverzeichnis:

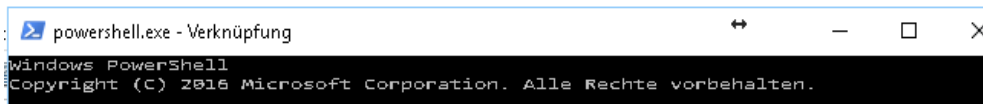


```
CA. Eingabeaufforderung
Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. Alle Rechte vorbehalten.
C:\Users\Franz-Josef>
```

Mit Vorteil wird das Programm `C:\Windows\Windows32\powershell.exe` in das Verzeichnis, in welchem Web-Anwendungen entwickelt werden sollen (`D:\Web`), kopiert und auf der Bemutzeroberfläche oder in der Taskleiste eine Verknüpfung darauf gespeichert.

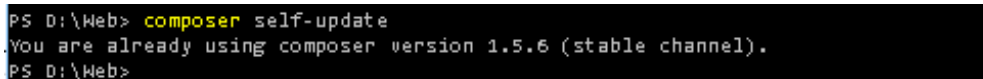


Ein Klick darauf startet dann die Eingabeaufforderung an der gewünschten Stelle:



```
powershell.exe - Verknüpfung
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. Alle Rechte vorbehalten.
```

Um sicherzustellen, dass lokal eine aktuelle Version von Composer vorhanden ist, führt man sinnvollerweise einen „self-update“ durch:



```
PS D:\Web> composer self-update
You are already using composer version 1.5.6 (stable channel).
PS D:\Web>
```

Auf die Abbildung der Befehle im Anwendungsfenster wird im folgenden Text verzichtet und nur noch die entsprechenden Befehle aufgeführt:

```
composer self-update
```

In der Dokumentation von **Laravel**¹² steht, dass ein Laravel-Projekt im Verzeichnis `D:\Web` wie folgt erstellt wird:

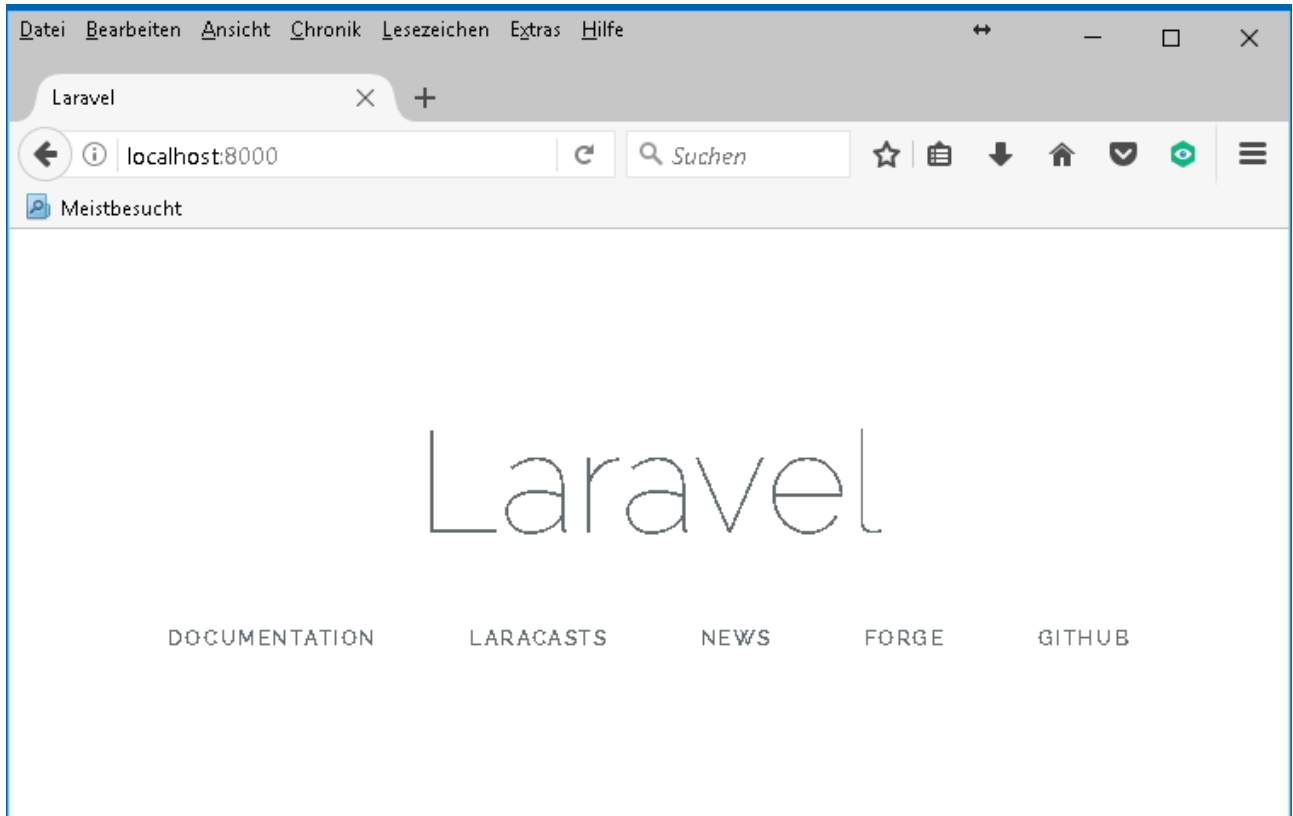
```
composer global require "laravel/installer"
laravel new MyWeb
```

¹² <https://laravel.com/docs>

Mit

```
D:\Web>CD MyWeb  
D:\Web\MyWeb>php artisan serve
```

wird in Verzeichnis MyWeb der wird der integrierte Web-Server gestartet. Zudem wird angegeben, dass die Website im Internet-Browser – zum Beispiel mit Firefox - mit dem Link <http://127.0.0.1:8000> oder <http://localhost:8000> respektive localhost:8000 aufgerufen werden kann.



Artisan ¹³ ist der Name eines mit Laravel gelieferten PHP-Programmes, welches eine Vielzahl Befehle ausführen kann, welche bei der Entwicklung einer Web-Anwendung nützliche Dienste leisten.

Um eine Website lokal zu entwickeln und auszutesten, braucht es eine Entwicklungs- und eine lokale Serverumgebung. Wie auf der Website Informatik von FJKarli.ch unter **Vorbereitung** ¹⁴ beschrieben, wird dazu **MAMP** ¹⁵ und **NetBeans IDE** ¹⁶ verwendet.

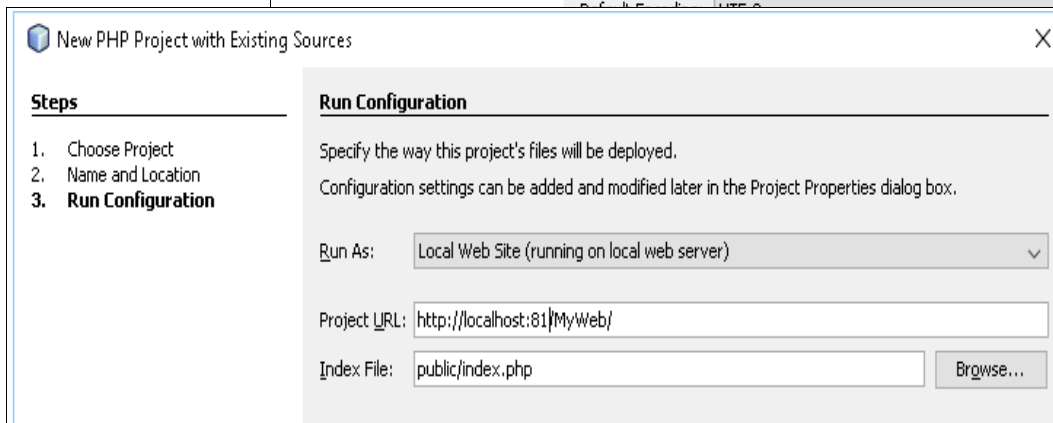
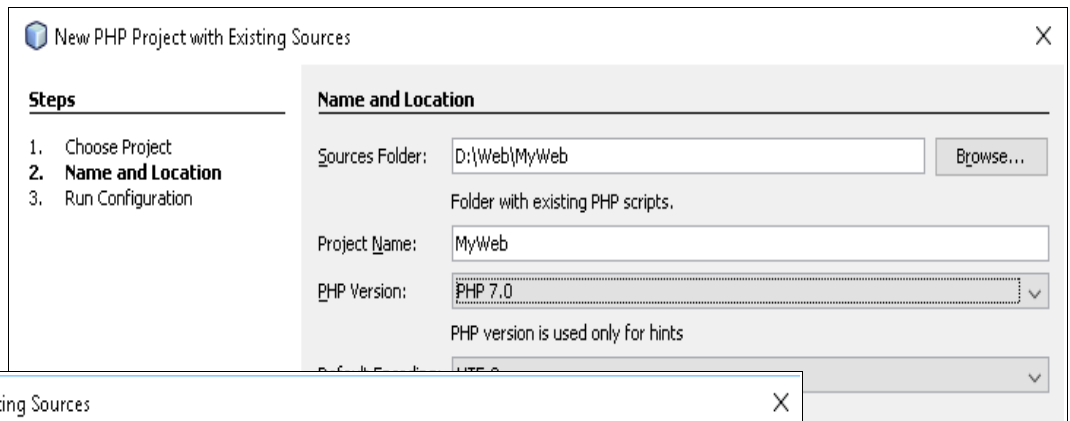
¹³ <https://laravel.com/docs/5.4/artisan>


¹⁴ <http://informatik.fjkarli.ch/MyWeb00.php>

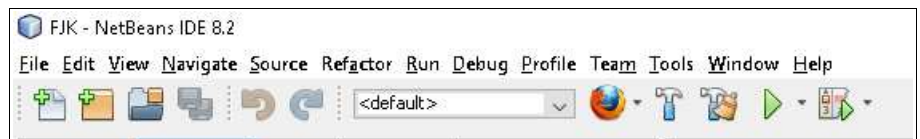
¹⁵ <http://www.mamp.info/de/> respektive <http://www.mamp.info/de/downloads/>

¹⁶ https://netbeans.org/index_de.html

Mit NetBeans wird ein neues PHP-Projekt erstellt: *File->New Project...->PHP Application with Existing Source*



Ein Klick auf  startet den Internet Browser



und liefert (mit Ausnahme) der URL das gleiche Resultat wie mit **php artisan serve**.

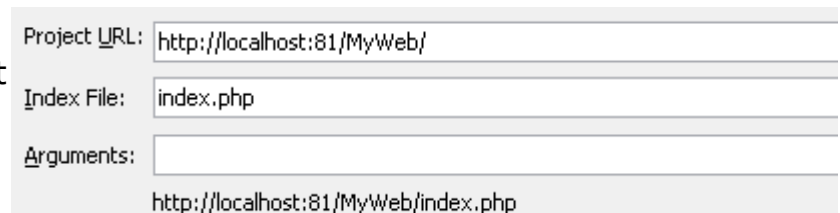
Eine Web-Anwendung macht nur Sinn, wenn sie auch auf einem Web-Server läuft. Zu diesem Zweck wurde (<http://www.MyWeb.FJKarli.ch>) erstellt.

Diese Webseite erwartet eine Datei index.html oder index.php im Stammverzeichnis. Laravel startet normalerweise eine Anwendung über die Datei public\index.php. Um die Anwendung auf einem „shared host“ laufen zu lassen, muss der Inhalt von public in das Stammverzeichnis kopiert und index.php wie folgt angepasst werden:

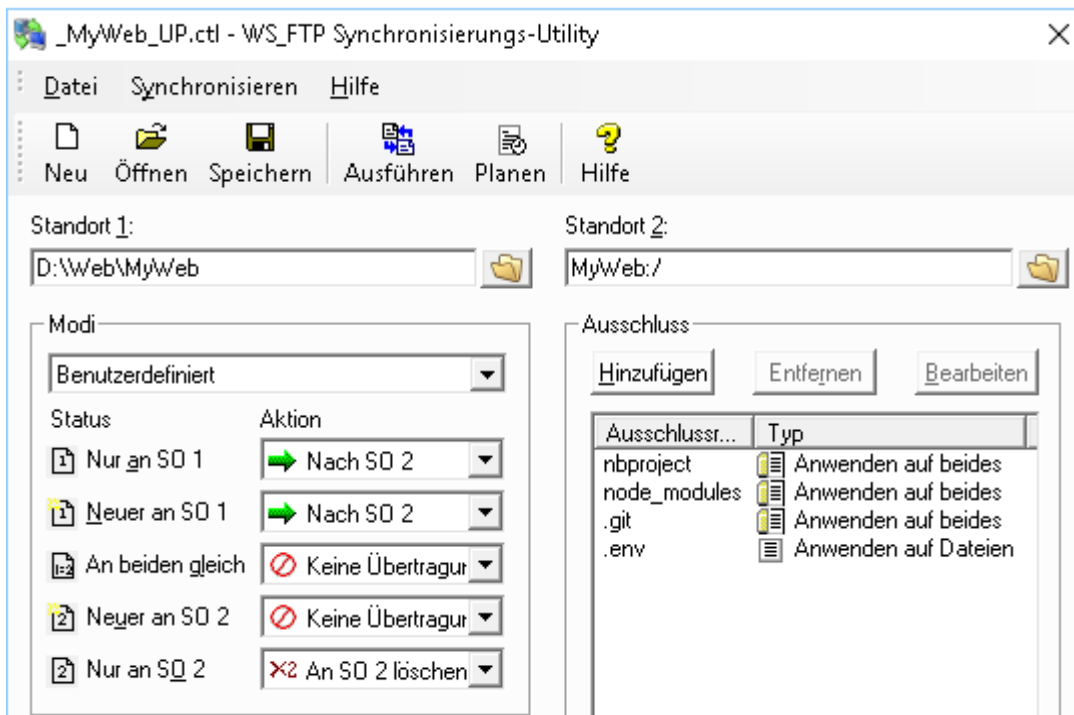
```
require __DIR__ . '/vendor/autoload.php';

$app = require_once __DIR__ . '/bootstrap/app.php';
```

In NetBeans muss der Link der neuen Situation angepasst werden.



Das Projekt kann nun (zum Beispiel mit WS_FTP Synchronisierungs-Utility) auf den Server kopiert werden.



Dabei ist zu beachten, dass gewisse Verzeichnisse und Dateien von der Synchronisation ausgeschlossen sind:

- nbproject
Wird ausschliesslich für die Entwicklungsumgebung NetBeans benutzt.
- node_modules
In diesem Verzeichnis werden mit npm (näheres später) Fremdpakete aus dem Internet gespeichert. Diese müssen dann in das Projekt kopiert werden. Der Inhalt von nodes_modules braucht es daher auf dem Web-Server nicht.
- .git
Arbeitet man mit einem Versionierungssystem, werden Änderungen im Verzeichnis .git gespeichert. Dieser Inhalt braucht es in der Anwendung auf dem Web-Server nicht.
- .env
Der Inhalt der Datei (mehr dazu später) unterscheidet sich auf dem Web-Server vom denjenigen der Entwicklungsumgebung. Die Datei .env auf dem Server darf daher bei der Aktualisierung der Web-Anwendung nicht überschrieben werden.

Der Aufruf von <http://myweb.fjkarli.ch/> liefert das gleiche Resultat wie die lokale Anwendung <http://localhost:81/MyWeb/index.php>.

Mit dieser kleinen Änderung läuft nun der Prototyp sowohl lokal als auch auf dem Web-Server. Die Datei `index.php` bereitet alles so vor, dass die Eingabe einer URL in einem Browser – beispielsweise <http://MyWeb.FJKarli.ch> - auf Grund der Angaben in der Datei `\routes\web.php` interpretiert und die entsprechende Seite aufruft. Die Startadresse der Website wird intern als '/' dargestellt.

Mit

```
\routes\web.php
<?php
Route::get('/', function () {
    return view('welcome');
});
```

wird angegeben, dass als Startadresse der Website die Datei welcome.blade.php ausgeführt werden soll. Laravel sucht Web-Seiten normalerweise unter `\resources\views` und Aufrufe im Browser beginnen meist mit der Startadresse. In Laravel können **markierte Angaben** weggelassen werden.

```
Route::get('http://MyWeb.FJKarli.ch/', function () {
    return view('\resources\views\welcome.blade.php');
});
```

Links respektive Routes verwenden den Präfix 'http://MyWeb.FJKarli.ch' und Views werden im Verzeichnis '`\resources\views\`' oder Unterverzeichnissen davon gespeichert.

In einigen Dateien verwendet Laravel einen sogenannten 'namespace'. Damit können in gewissen Fällen Pfadangaben vereinfacht werden.

Mit

```
php artisan app:name MyWeb
```

kann der 'namespace' App mit MyWeb ersetzt werden.