

## Vorbereitungen/Voraussetzungen für PHP Laravel Projekt

Bernd Engels, PHP Dozent

<https://laravel.com/docs/master>



Ziel ist die erstmalige Installation eines Laravel-Projekts, das unter der URL <http://videostore.loc> lokal zu Laufen gebracht werden soll, d.h. mittels eines Webbrowsers aufrufbar ist, um die eigene Arbeit am Projekt ständig darüber testen zu können.

Benötigt wird :

- Apache als Webserver
- PHP als Apache-Modul
- MySQL-Datenbank Server
- Mercury Mail-Server (Windows) / Postfix oder anderer Mail-Server (Linux, MacOS).

Unter Windows reicht hier die Installation Von XAMPP. Darin ist alles Notwendige enthalten.

<https://www.apachefriends.org/de/index.html>

Im Xampp Verzeichnis klicken wir 'xampp-control.exe' an und starten hier den Apache-, MySQL- und Mercury-Server. Wenn das erfolgt ist, sollten diese Dienste bei jedem Neustart des Rechners automatisch starten. Da wir bei allen Konfigurationen dieser Services, die entsprechenden Server neu starten müssen, ist es ratsam, sich auf den Desktop eine Verknüpfung zu xampp-control.exe anzulegen.

Für die Arbeit an Konfigurations-Dateien von Servern ist es ratsam, sich eine guten Texteditor anzulegen, der in der Lage ist, auch Zeilennummern anzuzeigen zwecks besserer Kommunikation untereinander (Was wo in welcher Zeile zu finden ist): TIP: **Textpad** <https://www.textpad.com/>

Darüber hinaus wird gebraucht:

- Composer (als PHP Paket Manager) <https://getcomposer.org/>
- Node.js (daraus 'npm' als Node Package Manager' für JS-Module im weitesten Sinne) <https://nodejs.org/en/>

Nach den Installationen (Download der Software vorausgesetzt), sollte auf der Kommando-Zeile des System Terminal (unter Windows cmd.exe) folgende beiden Kommandos ohne Fehlermeldung aufrufbar sein:

– composer

```
Last login: Wed Nov  6 18:33:29 on ttys002
[+] ~ composer

Composer version 1.8.2 2019-01-29 15:00:53

Usage:
  command [options] [arguments]

Options:
  -h, --help                Display this help message
  -q, --quiet               Do not output any message
  -V, --version             Display this application version
  --ansi                    Force ANSI output
  --no-ansi                 Disable ANSI output
  -n, --no-interaction      Do not ask any interactive question
  --profile                 Display timing and memory usage information
  --no-plugins              Whether to disable plugins.
  -d, --working-dir=WORKING-DIR If specified, use the given directory as working directory.
  -v|vv|vvv, --verbose      Increase the verbosity of messages: 1 for normal output, 2 for more ve

Available commands:
  about      Shows the short information about Composer.
```

– 'npm

```
[+] ~ npm

Usage: npm <command>

where <command> is one of:
  access, adduser, audit, bin, bugs, c, cache, ci, cit,
  clean-install, clean-install-test, completion, config,
  create, ddp, dedupe, deprecate, dist-tag, docs, doctor,
  edit, explore, get, help, help-search, hook, i, init,
  install, install-ci-test, install-test, it, link, list, ln,
  login, logout, ls, org, outdated, owner, pack, ping, prefix,
  profile, prune, publish, rb, rebuild, repo, restart, root,
  run, run-script, s, se, search, set, shrinkwrap, star,
  stars, start, stop, t, team, test, token, tst, un,
  uninstall, unpublish, unstar, up, update, v, version, view,
  whoami

npm <command> -h  quick help on <command>
npm -l            display full usage info
npm help <term>   search for help on <term>
npm help npm      involved overview

Specify configs in the ini-formatted file:
  /Users/bengels/.npmrc
or on the command line via: npm <command> --key value
Config info can be viewed via: npm help config

npm@6.12.1 /usr/local/lib/node_modules/npm
➔ ~
```

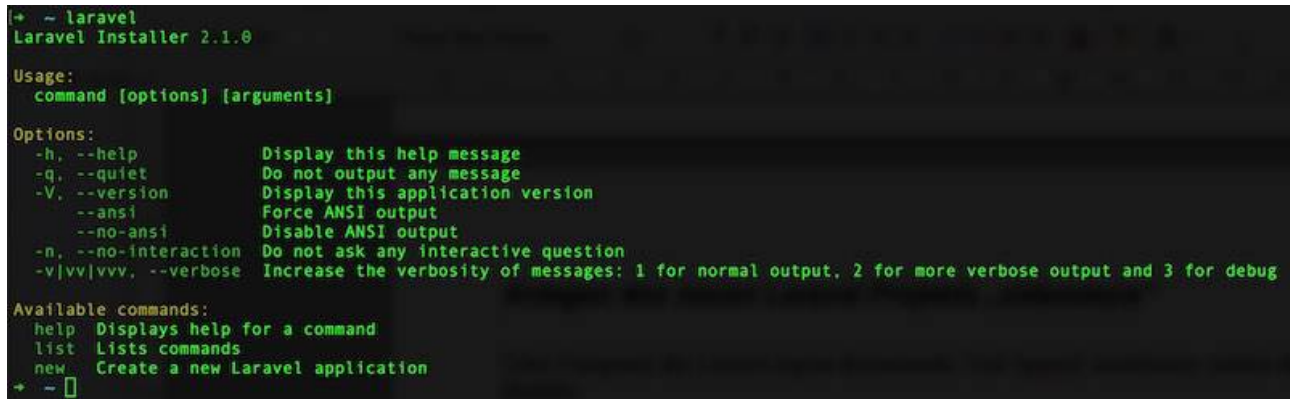
Unter Windows sollte hier nach der Installation ein Neustart erforderlich sein, da die Anpassungen an der Umgebungsvariable 'PATH', die bei den Installationen erfolgen, erst nach diesem wirksam werden.

## Anlegen des neuen Laravel Projekts „videostore“

Über Composer das Laravel eigene Kommando Tool 'laravel' installieren, mittels des Konsole-Befehls:

```
composer global require laravel/installer
```

Danach sollte das 'laravel'-Kommando auf der Befehlszeile der Terminals aufrufbar sein:



```
[+ ~ laravel
Laravel Installer 2.1.0

Usage:
  command [options] [arguments]

Options:
  -h, --help            Display this help message
  -q, --quiet           Do not output any message
  -V, --version         Display this application version
      --ansi            Force ANSI output
      --no-ansi        Disable ANSI output
  -n, --no-interaction Do not ask any interactive question
  -v|vv|vvv, --verbose Increase the verbosity of messages: 1 for normal output, 2 for more verbose output and 3 for debug

Available commands:
  help  Displays help for a command
  list  Lists commands
  new   Create a new Laravel application
+ ~ ]
```

Wir bleiben im Terminal (cmd.exe) und wechseln nun in das Hauptverzeichnis der Web-Dokumente des Apache Web-Servers. (Bei XAMPP liegt dieses standardmäßig unter: C:\xampp\htdocs).

```
cd c:\xampp\htdocs
```

Danach legen wir hier das neue Projekt an mittels:

```
laravel new videostore
```

Laravel legt jetzt hier ein gleichnamiges Verzeichnis mit den Projektdaten an. Wir öffnen jetzt dieses Verzeichnis mit dem normalen Datei-Explorer und nennen die darin enthaltene Datei 'index.php' um in 'index.php.bak'.

Wenn wir jetzt mit dem Webbrowser die Adresse: <http://localhost> öffnen, sollte eine Liste aller Dateien und Verzeichnisse des Ordners 'c:\xampp\htdocs' erscheinen (inkl. unseres videostore Verzeichnisses).

Um das neue Projekt im Browser testen zu können, genügt es nicht, sich einfach über den Browser dort hinein zu klicken. Wir müssen hierzu eine sog. Virtuellen Host über den Apache einrichten und darüber hinaus eine Anpassung in der Host-Datei des Windows-Rechners. Diese beiden Dateien finden wir unter:

- C:\xampp\apache\conf\extra\httpd-vhost.conf
- C:\Windows\System32\drivers\etc\hosts

In C:\Windows\System32\drivers\etc\hosts mittels eines reinen Texteditors (Textpad) folgendes eintragen (Wichtig: Die Datei-Rechte so anpassen, daß wir sie ändern dürfen):

```
127.0.0.1    localhost
::1         localhost
127.0.0.1    videostore.loc
127.0.0.1    admin.videostore.loc
```

Das dient dazu, daß das Betriebssystem (Windows) unabhängig vom Apache Server die darin enthaltenen Domain-Namen der lokalen IP-Adresse zuordnet. Wenn wir das nicht tun würden, würde das System davon ausgehen, daß es sich um reale Web-Adressen handelt.

Als nächstes passen wir die Vhost-Datei des Apache an (C:\xampp\apache\conf\extra\httpd-vhost.conf) und tragen darin folgendes ein:

```
NameVirtualHost *:80

# localhost
<VirtualHost *:80>
    DocumentRoot "C:/xampp/htdocs"
    ServerName localhost
    ServerAlias local
    ErrorLog "logs/local-error.log"
    CustomLog "logs/local-access.log" common
</VirtualHost>

# videostore.loc
<VirtualHost *:80>
    DocumentRoot "C:/xampp/htdocs/videostore/public"
    ServerName videostore.loc
    ServerAlias admin.videostore.loc
    ErrorLog "logs/videostore-error.log"
    CustomLog "logs/videostore-access.log" common
</VirtualHost>
```

Anschliessend starten wir über das XAMPP-Control-Panel den Apache Server neu, damit die Änderungen wirksam werden. Zum Testen öffnen wir jetzt über den Web-Browser unsere neue Projekt URL 'videostore.loc' und sollten nun die von Laravel vorgegebene Welcome Page sehen.

Als letzten Schritt legen wir in unser videostore Verzeichnis eine Datei namens .htaccess an (wichtig der Punkt am Anfang des Namens) und tragen darin folgende Zeile ein:

```
RedirectPermanent /videostore http://videostore.loc
```

Das bewirkt, daß beim Klick auf das videostore-Verzeichnisses im Webbrowser, sich sofort die Webseite '[\*\*http://videostore.loc\*\*](http://videostore.loc)' öffnet und wir nicht einfach in dem Unterverzeichnis landen.