# Laravel 6

*Notes open for creative commons use @ developer blog: https://unfoldkyle.com, github:* SmilingStallman*, email: kmiskell@protonmail.com*

**Learning Resources**
Primary: Official Docs, https://laravel.com/docs/6.x
Secondary: https://laracasts.com/

## Intro

-PHP MVC framework based on Symfony with version 1.0 in 2011, and current version (6.13)

-Built in authentication, restful routing, lightweight templating through Blade templating engine, unit testing supported out of box, wide array of OO libraries built in.

-Built in command line interface through Artisan

-Highly modularized packaging system with dedicated dependency manager

-Prepare SQL statements only (preventing sql injection), as well as many other security features

-General flow: User submits request for page. Laravel sees request and routes to proper controller based on config in *routes.php*. Controller, which holds domain knowledge and business logic, gets request. Back-end logic runs, controller passes data to view, then view loads and is displayed to user.

-Note that somePage.php and somePageController.php are separated, so controller acts as de-coupled between view and back-end logic.

-Laravel 6 requires PHP 7.2 (released Nov 2017) or higher

-Using a framework like laravel lets you not re-invent the wheel, handles security well by default, can provide performance updates, and makes extension easy via plugins. It also standardizes code when used in multi-projects, files, etc. by providing a common design and implementation.

**Installation**
-Install PHP, MySQL, Composer, Laravel

https://www.digitalocean.com/community/tutorials/how-to-install-linux-nginx-mariadb-php-lemp-stack-on-debian-10

https://getcomposer.org/download/

https://laravel.com/docs/6.x#installation https://laracasts.com/series/laravel-6-from-scratch/episodes/3

-If Laravel install complains about missing ext-zip missing, install via:
*sudo apt-get install php-zip*

-On Debian, set *$PATH* by adding below to *~/.bashrc*
*export PATH="$HOME/.config/composer/vendor/bin:$PATH"*

### Why Use in Building API

-MVC works well for building properly designed API
-Eloquent ORM through Artisan allows for auto-creation of models based on table existing scheme and allows proper DB layer
-Automated unit test
-Fast, simple, clean routing
-Managed queuing during high load times to increase speed
-Easy and quick autoloading

### Composer, Namespaces, and Autoloading

-Knowledge on all of these will be required for aptly using Laravel. See *PHP7.odt* notes for info on these under sections, "Advanced PHP," and, "Composer"

### New Project

-Create new laravel project: *laravel new projectName*

-On first attempt to create new on Debian, PHP was missing dependencies. Install via:
*sudo apt-get install php-mbstring*
*sudo apt-get install php-dom*

-Creation will create folder *projectName* with all required dependencies, as well as template html page, similar to React via *create-react-app*

-Laravel comes with config for local dev server for project, typically running on *http://127.0.0.1:8000*. From project dir, start via: *php artisan serve*

### Webserver Project Config

-Root directory for webserver should be set to *my_project/public* which holds *index.php,* which starts Laravel, includes autoloading for dependencies, requires Bootstrap, and serves as front controller for all HTTP requests entering app

-Ensure project *storage* and *bootstrap/cache* dir have write permissions

-If did not build project with *laravel new* (aka composer), set app key for security modules via:
*php artisan key:generate*

-Additional useful config (ex. timezone) in *my_project/config/app.php*

-Depending on project, also might want to configure *my_project/config/cache.php, my_project/config/database.php,* and *my_project/config/session.php*

-For security reasons, always host laravel projects in the root folder for the “web directory” ( ex. */var/www/* ). Example:

```
server {
        listen 80;
        listen [::]:80;

        root /var/www/html/quickstart/public;
        index index.php index.html index.htm index.nginx-debian.html;

        server_name example.com www.example.com;

        location / {
            try_files $uri $uri/ /index.php?$query_string;
        }
}
```

For more details on deployment:
https://www.digitalocean.com/community/tutorials/how-to-deploy-a-laravel-application-with-nginx-on-ubuntu-16-04