Laravel Introduction

Set up

You are going to set up a Laravel project in your *Documents* folder using Composer.

Get Composer

Laravel uses Composer to manage its dependencies.

If you have not already got a copy of *composer.phar* on your computer you will need to download it from https://getcomposer.org/download/.

Scroll down to the “Manual Download” section and click on “Latest Snapshot”. Download *composer.phar* to your *Documents* folder.

Open a command-prompt

Navigate to your *Documents* folder. Enter *cmd* into the address bar and press enter. This should open a command-prompt at your *Documents* folder.

Set PHP in your path

Assuming a standard *xampp* installation on a *Windows* machine, you will need to call *C:|xampp|php|php.exe* a number of times from the command-prompt. To save you from having to keep typing this you can add it to your path for the current command-prompt window. If you open another command-prompt window you will have to add it again for that window.

```
set PATH=%PATH%;C:\xampp\php
```

Create a Laravel project

Install Laravel

Make sure that your command-prompt is set to your *Documents* folder and (assuming that composer.phar is in your documents folder) type the following:

```
php composer.phar require Laravel/installer
```

There are alternative ways in which you can install Laravel for your projects. For more information on installation visit https://laravel.com/docs/6.x.

Create a new project

Type the following at the command prompt:

```
php composer.phar create-project laravel/laravel laravel-tutorial
```

Composer will download everything you need for your Laravel project. This may take a few minutes to complete.

For information,

- ***create-project***[1] is a composer command
- ***laravel/laravel*** refers to the package used for the Laravel installation
- ***laravel-tutorial*** is the name of the project you want to create.

You can also include a Laravel version number after your project name if you want to specify the version of Laravel that you wish to use.

A project directory called *laravel-tutorial* should now have been created within your *Documents* folder.

View your project in a browser

Navigate to your project directory.

```
cd laravel-tutorial
```

[1] https://getcomposer.org/doc/03-cli.md#create-project

Laravel includes a command-line interface called *artisan* which gives you access to a host of useful commands to make developing projects much easier.

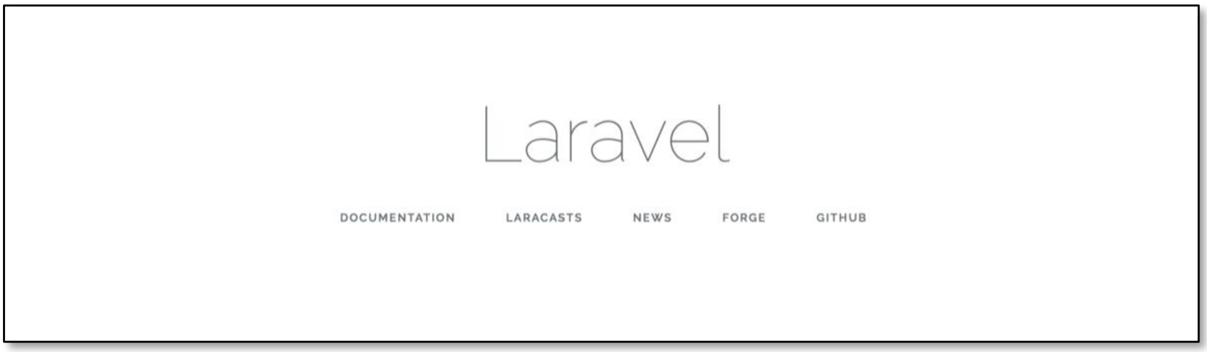
To start a local web server so that you can view your project use the following command:

```
php artisan serve
```

You should get the following response:

```
Laravel development server started: <http://127.0.0.1:8000>
```

View http://127.0.0.1:8000.



You can regain the command-prompt and stop the server using Ctrl+C.

The project directory structure

```
├── app
│   ├── Console
│   ├── Exceptions
│   ├── Http
│   │   ├── Controllers
│   │   └── Middleware
│   └── Providers
├── bootstrap
├── config
├── database
│   ├── factories
│   ├── migrations
│   └── seeds
├── public
├── resources
│   ├── js
│   ├── sass
│   ├── lang
│   └── views
├── routes
├── storage
├── tests
└── vendor
```

This is the directory structure to a maximum of two levels with no files shown. Directories have been highlighted where they might be familiar from the rudimentary framework we created during the MVC From Scratch tutorials.

In addition:

app contains core code for the application – it also contains a *Controllers* folder where your controller classes will go,

config has default configuration options for the project,

public contains the controlling *index.php* and publically available assets such as *CSS* files and *JavaScript*,

resources contains your *views* in the *views* folder alongside localisation files and pre-compiled assets,

tests contains the automated tests for your project.