

Prologue

Getting Started

Architecture Concepts

• Request Lifecycle

Service Container

Service Providers

Facades

Contracts

The Basics

Frontend

Security

Digging Deeper

Database

Eloquent ORM

Testing

Official Packages



Deploy Laravel and PHP applications on DigitalOcean, Linode, & AWS.

ADS VIA CARBON

# Request Lifecycle

# Introduction

# Lifecycle Overview

# Focus On Service Providers

## # Introduction

When using any tool in the "real world", you feel more confident if you understand how that tool works. Application development is no different. When you understand how your development tools function, you feel more comfortable and confident using them.

The goal of this document is to give you a good, high-level overview of how the Laravel framework works. By getting to know the overall framework better, everything feels less "magical" and you will be more confident building your applications. If you don't understand all of the terms right away, don't lose heart! Just try to get a basic grasp of what is going on, and your knowledge will grow as you explore other sections of the documentation.

## # Lifecycle Overview

### First Things

The entry point for all requests to a Laravel application is the `public/index.php` file. All requests are directed to this file by your web server (Apache / Nginx) configuration. The `index.php` file doesn't contain much code. Rather, it is a starting point for loading the rest of the framework.

The `index.php` file loads the Composer generated autoloader definition, and then retrieves an instance of the Laravel application from `bootstrap/app.php` script. The first action taken by Laravel itself is to create an instance of the application / [service container](#).

### HTTP / Console Kernels

Next, the incoming request is sent to either the HTTP kernel or the console kernel, depending on the type of request that is entering the application. These two kernels serve as the central location that all requests flow through. For now, let's just focus on the HTTP kernel, which is located in `app/Http/Kernel.php`.

The HTTP kernel extends the `Illuminate\Foundation\Http\Kernel` class, which defines an array of `bootstrappers` that will be run before the request is executed. These bootstrappers configure error handling, configure logging, [detect the application environment](#), and perform other tasks that need to be done before the request is actually handled.

The HTTP kernel also defines a list of HTTP [middleware](#) that all requests must pass through before being handled by the application. These middleware handle reading and writing the [HTTP session](#), determining if the application is in maintenance mode, [verifying the CSRF token](#), and more.

The method signature for the HTTP kernel's `handle` method is quite simple: receive a `Request` and return a `Response`. Think of the Kernel as being a big black box that represents your entire application. Feed it HTTP requests and it will return HTTP responses.

### Service Providers

One of the most important Kernel bootstrapping actions is loading the [service providers](#) for your application. All of the service providers for the application are configured in the `config/app.php` configuration file's `providers` array. First, the `register` method will be called on all providers, then, once all providers have been registered, the `boot` method will be called.

Service providers are responsible for bootstrapping all of the framework's various components, such as the database, queue, validation, and routing components. Since they bootstrap and configure every feature offered by the framework, service providers are the most important aspect of the entire Laravel bootstrap process.

### Dispatch Request

Once the application has been bootstrapped and all service providers have been registered, the `Request` will be handed off to the router for dispatching. The router will

dispatch the request to a route or controller, as well as run any route specific middleware.

## # Focus On Service Providers

Service providers are truly the key to bootstrapping a Laravel application. The application instance is created, the service providers are registered, and the request is handed to the bootstrapped application. It's really that simple!

Having a firm grasp of how a Laravel application is built and bootstrapped via service providers is very valuable. Your application's default service providers are stored in the `app/Providers` directory.

By default, the `AppServiceProvider` is fairly empty. This provider is a great place to add your application's own bootstrapping and service container bindings. For large applications, you may wish to create several service providers, each with a more granular type of bootstrapping.

## Become a Laravel Partner

Laravel Partners are elite shops providing top-notch Laravel development and consulting. Each of our partners can help you craft a beautiful, well-architected project.

Our Partners

# Laravel

## Highlights

[Release Notes](#)  
[Getting Started](#)  
[Routing](#)  
[Blade Templates](#)  
[Authentication](#)  
[Authorization](#)  
[Artisan Console](#)  
[Database](#)  
[Eloquent ORM](#)  
[Testing](#)

## Resources

[Laracasts](#)  
[Laravel News](#)  
[Laracon](#)  
[Laracon EU](#)  
[Laracon AU](#)  
[Jobs](#)  
[Certification](#)  
[Forums](#)

## Partners

[Vehikl](#)  
[Tighten Co.](#)  
[Kirschbaum](#)  
[Byte 5](#)  
[64Robots](#)  
[Cubet](#)  
[DevSquad](#)  
[Ideil](#)  
[Cyber-Duck](#)  
[ABOUT YOU](#)  
[Become A Partner](#)

## Ecosystem

[Vapor](#)  
[Forge](#)  
[Envoyer](#)  
[Horizon](#)  
[Lumen](#)  
[Nova](#)  
[Echo](#)  
[Valet](#)  
[Mix](#)  
[Spark](#)  
[Cashier](#)  
[Homestead](#)  
[Dusk](#)  
[Passport](#)  
[Scout](#)  
[Socialite](#)

Laravel is a web application framework with expressive, elegant syntax. We believe development must be an enjoyable and creative experience to be truly fulfilling. Laravel attempts to take the pain out of development by easing common tasks used in most web projects.

Laravel is a Trademark of Taylor Otwell.  
Copyright © 2011-2019 Laravel LLC.

