Laravel – ett fullskaligt MVC-ramverk

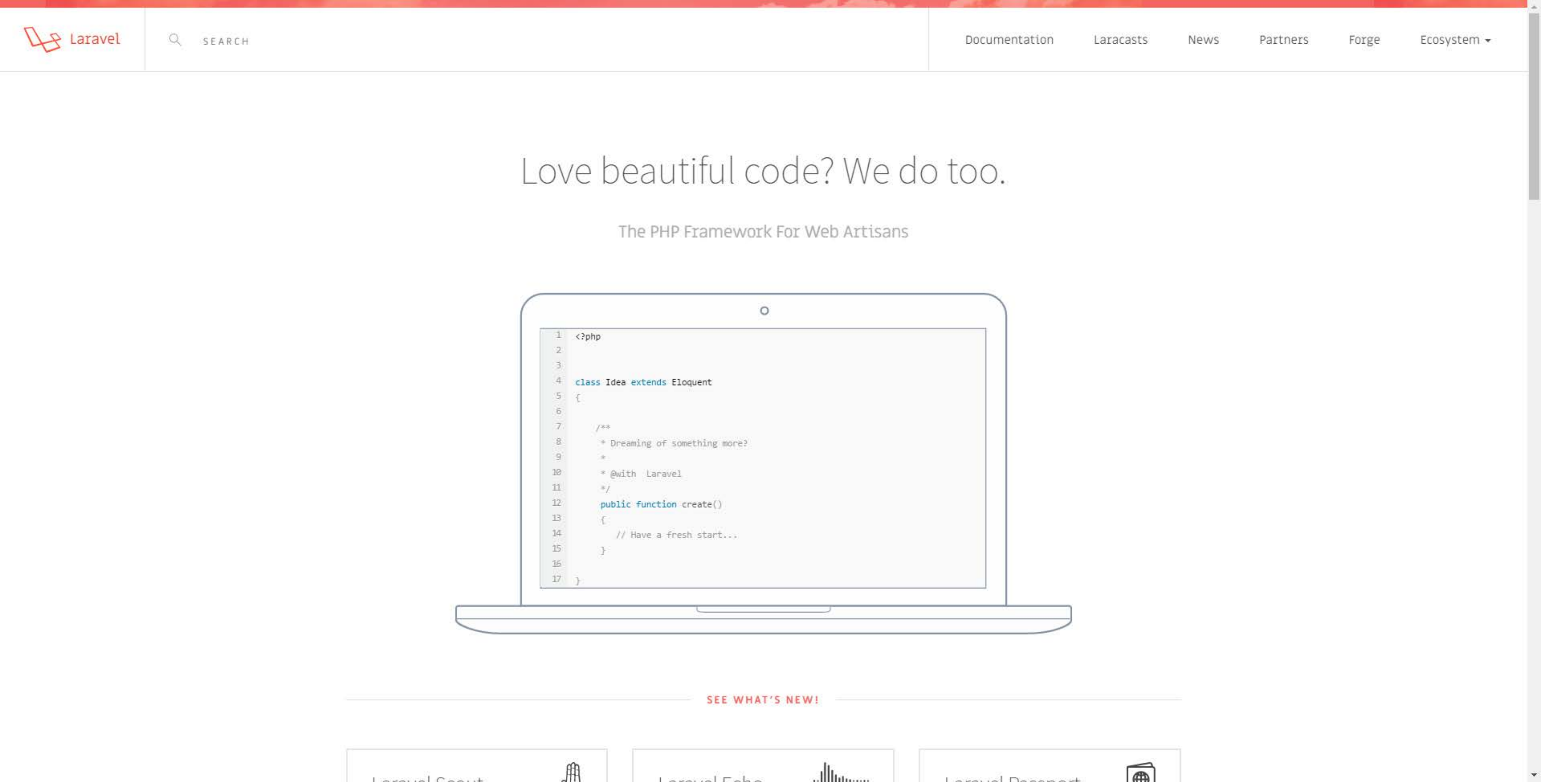
Fortsättning…

Dagens agenda

- Snabb repetition
- Middlewares
- Användarhantering i Laravel
  - Användardatabas
  - Inloggning
- Summering av denna del i kursen

Repetetion

Laravel med tillhörande koncept



Love beautiful code? We do too.

The PHP Framework For Web Artisans

```
<?php


class Idea extends Eloquent
{

    /**
     * Dreaming of something more?
     *
     * @with  Laravel
     */
     public function create()
     {
        // Have a fresh start...
     }

}
```

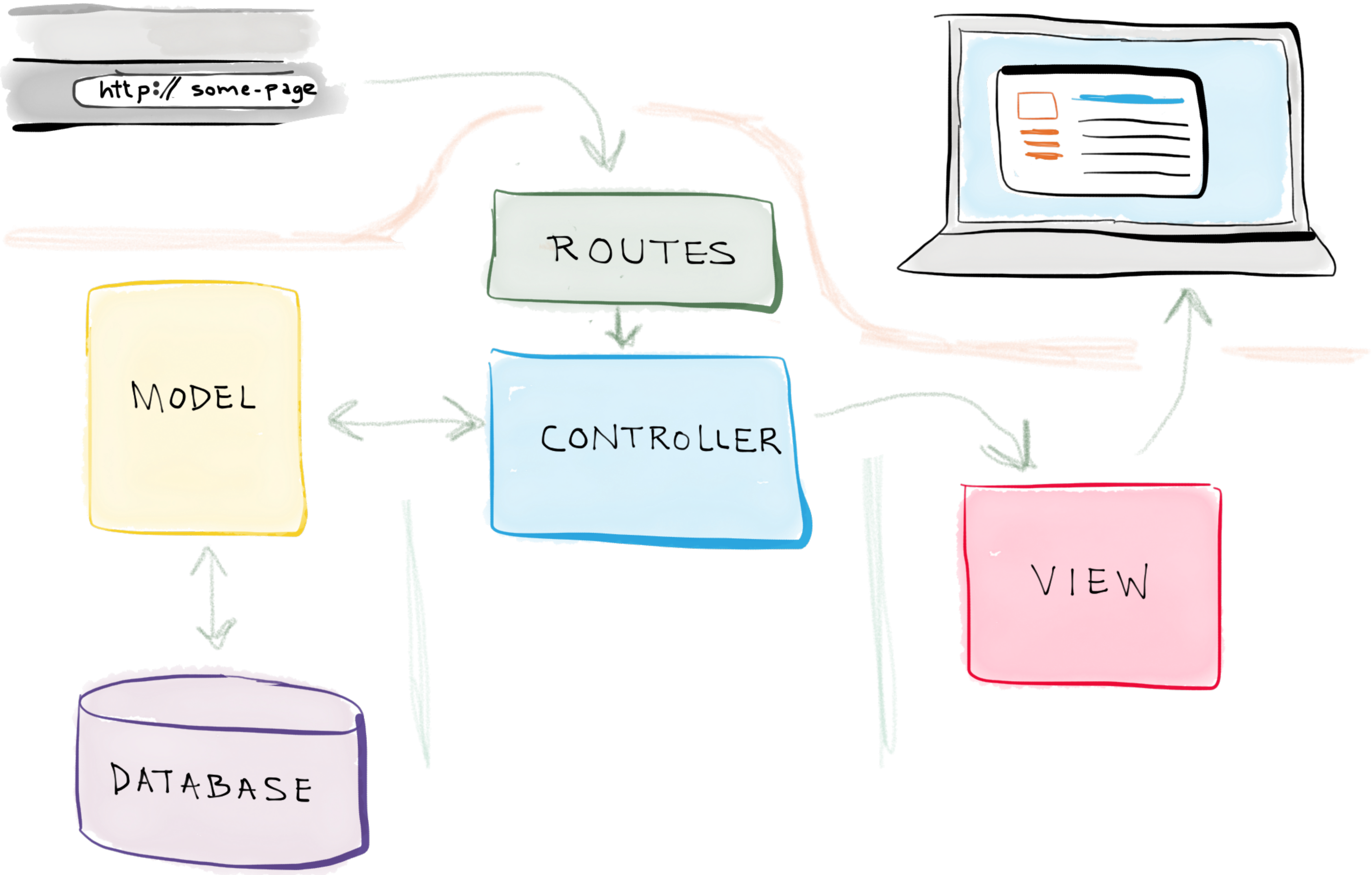
SEE WHAT'S NEW!

Laravel Scout

Laravel Echo

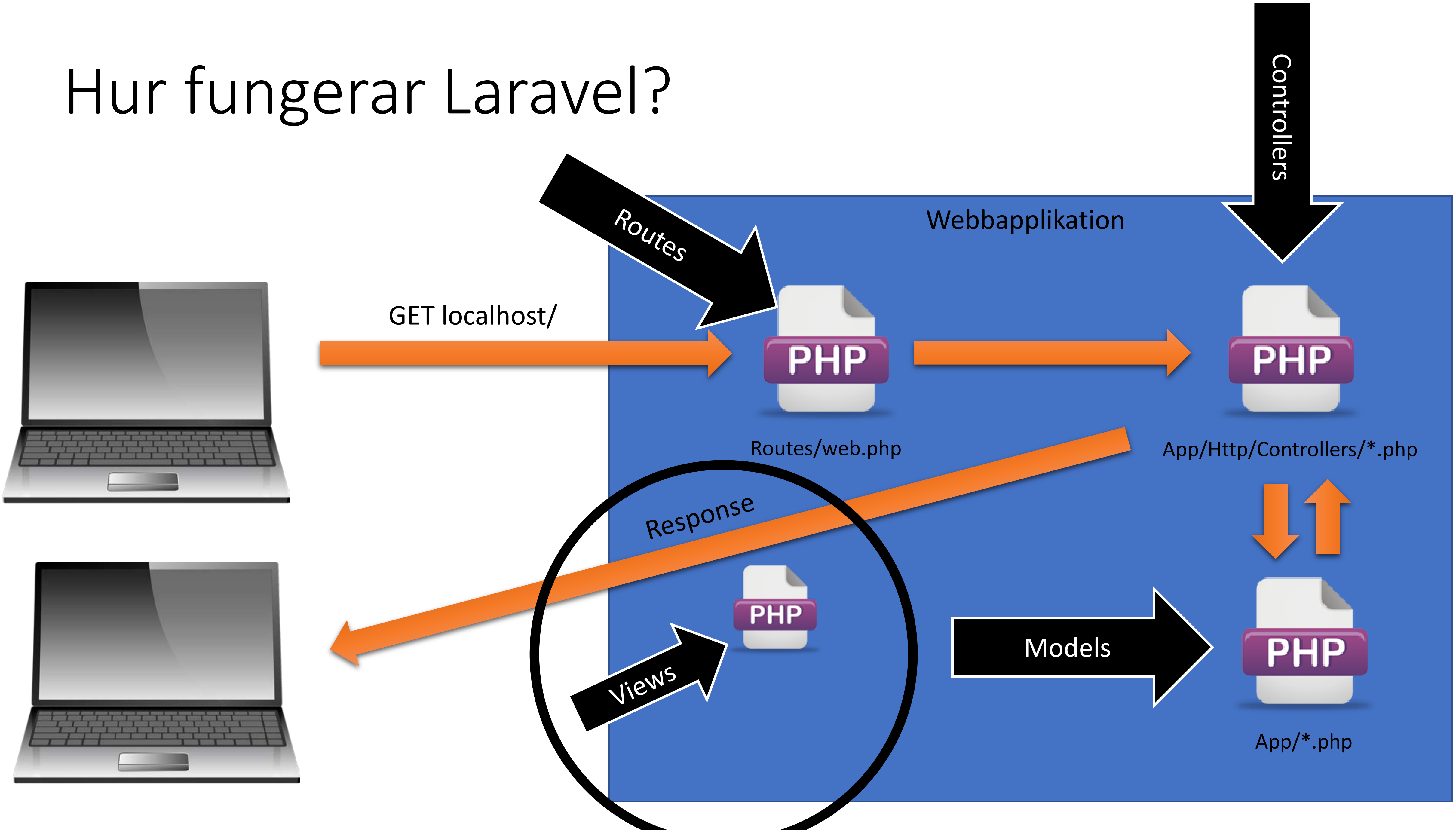
Laravel Passport

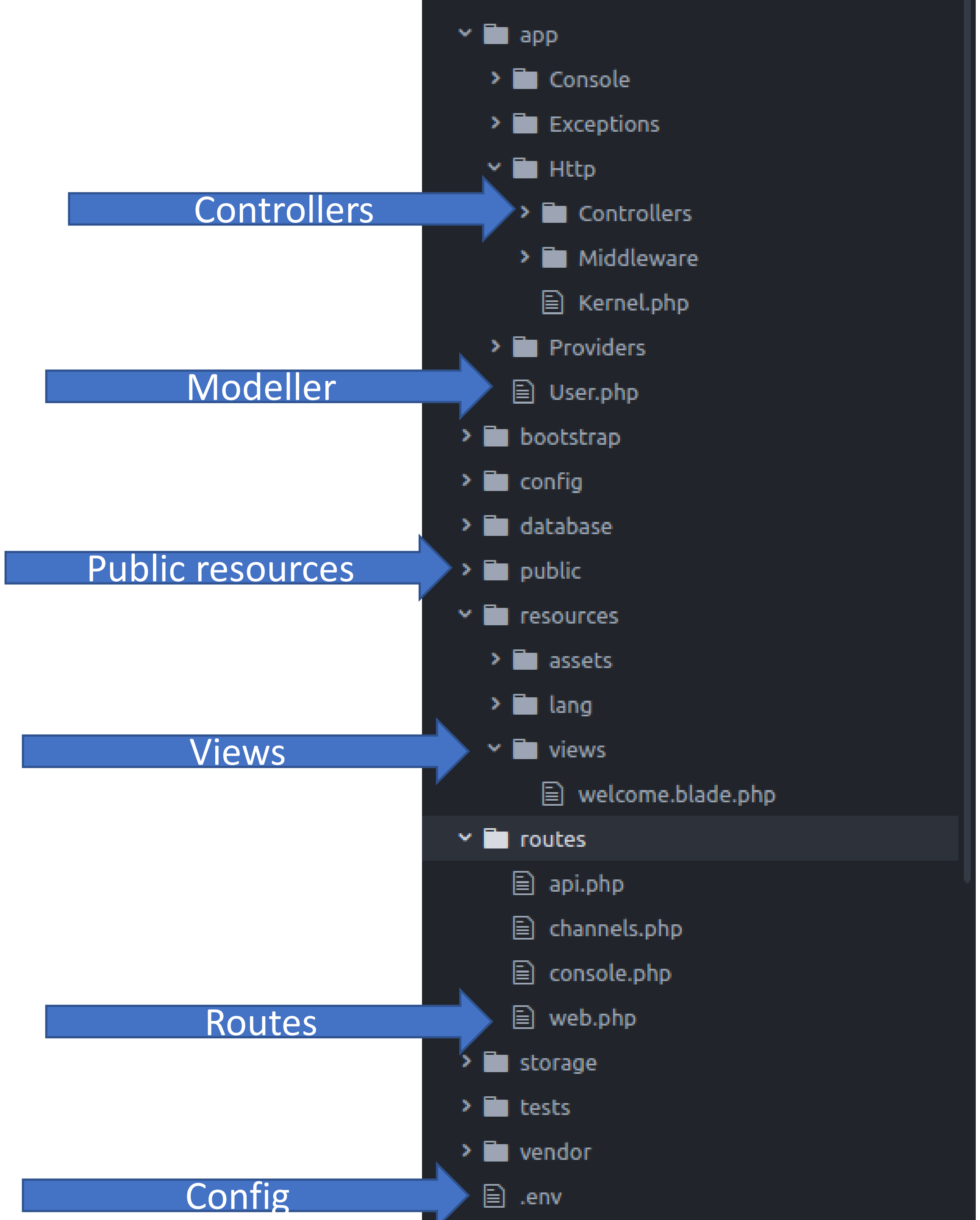
https://laravel.com/



https://selftaughtcoders.com/wp-content/uploads/2015/07/mvc_diagram_with_routes.png

Hur fungerar Laravel?





Controllers

Modeller

Public resources

Views

Routes

Config

```
app
  Console
  Exceptions
  Http
    Controllers
    Middleware
    Kernel.php
  Providers
  User.php
bootstrap
config
database
public
resources
  assets
  lang
  views
    welcome.blade.php
routes
  api.php
  channels.php
  console.php
  web.php
storage
tests
vendor
.env
```

Lumen =>

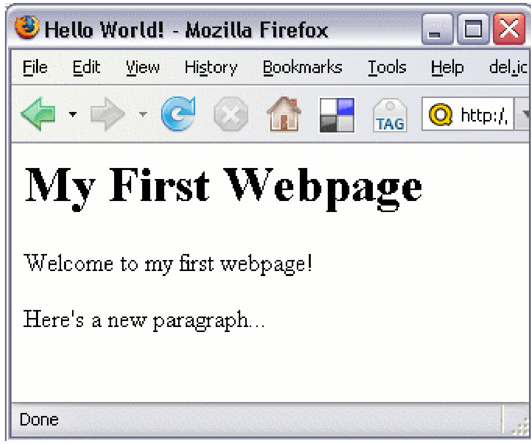
Routes
Controllers
Models
Database

=> JSON-data

Laravel =>

Routes
Controllers
Models
Database

=> Webbsida

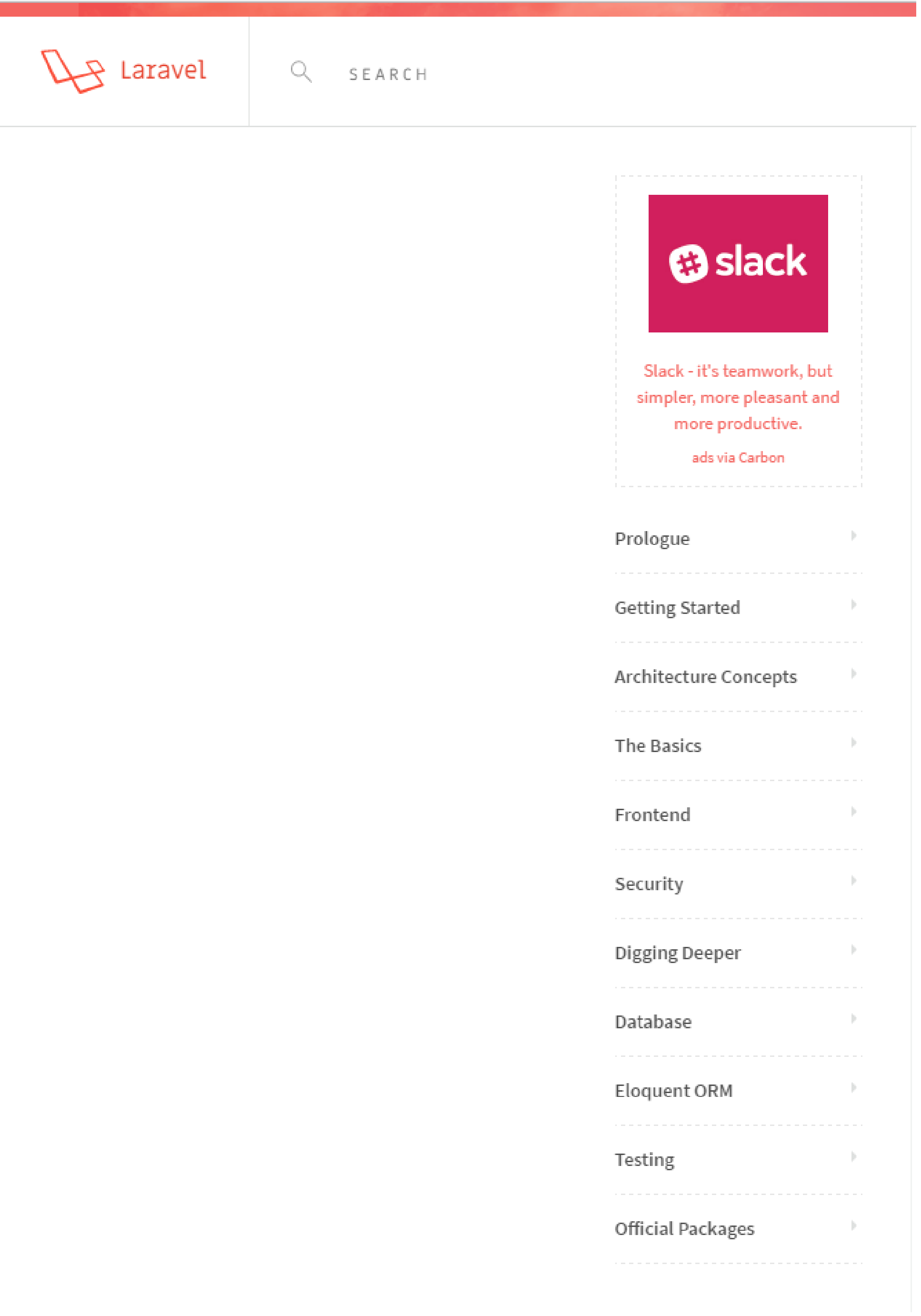

Hello World! - Mozilla Firefox
File Edit View History Bookmarks Tools Help del.ic
TAG
http:/,

My First Webpage

Welcome to my first webpage!

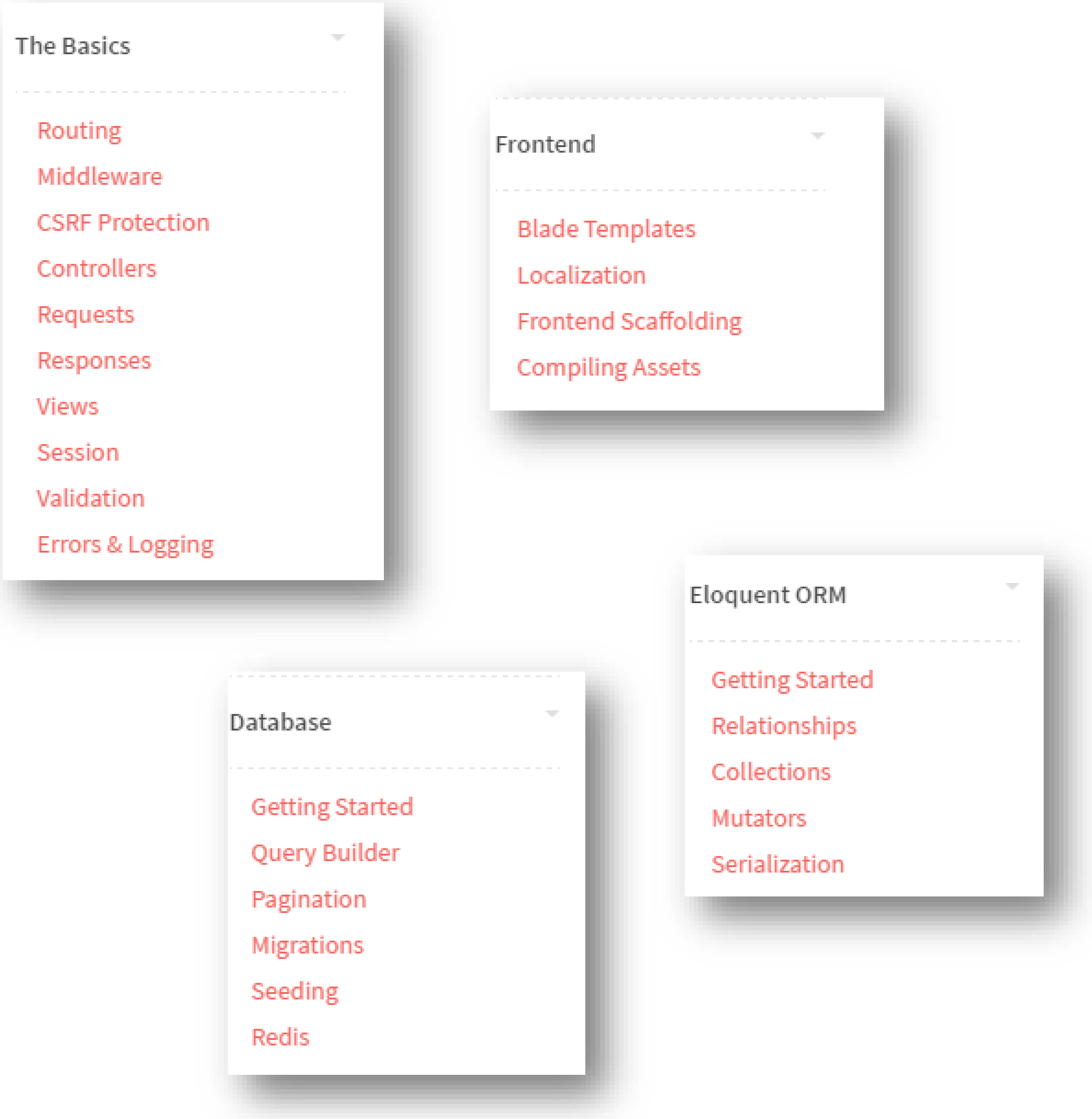
Here's a new paragraph...

Done



Laravel
SEARCH

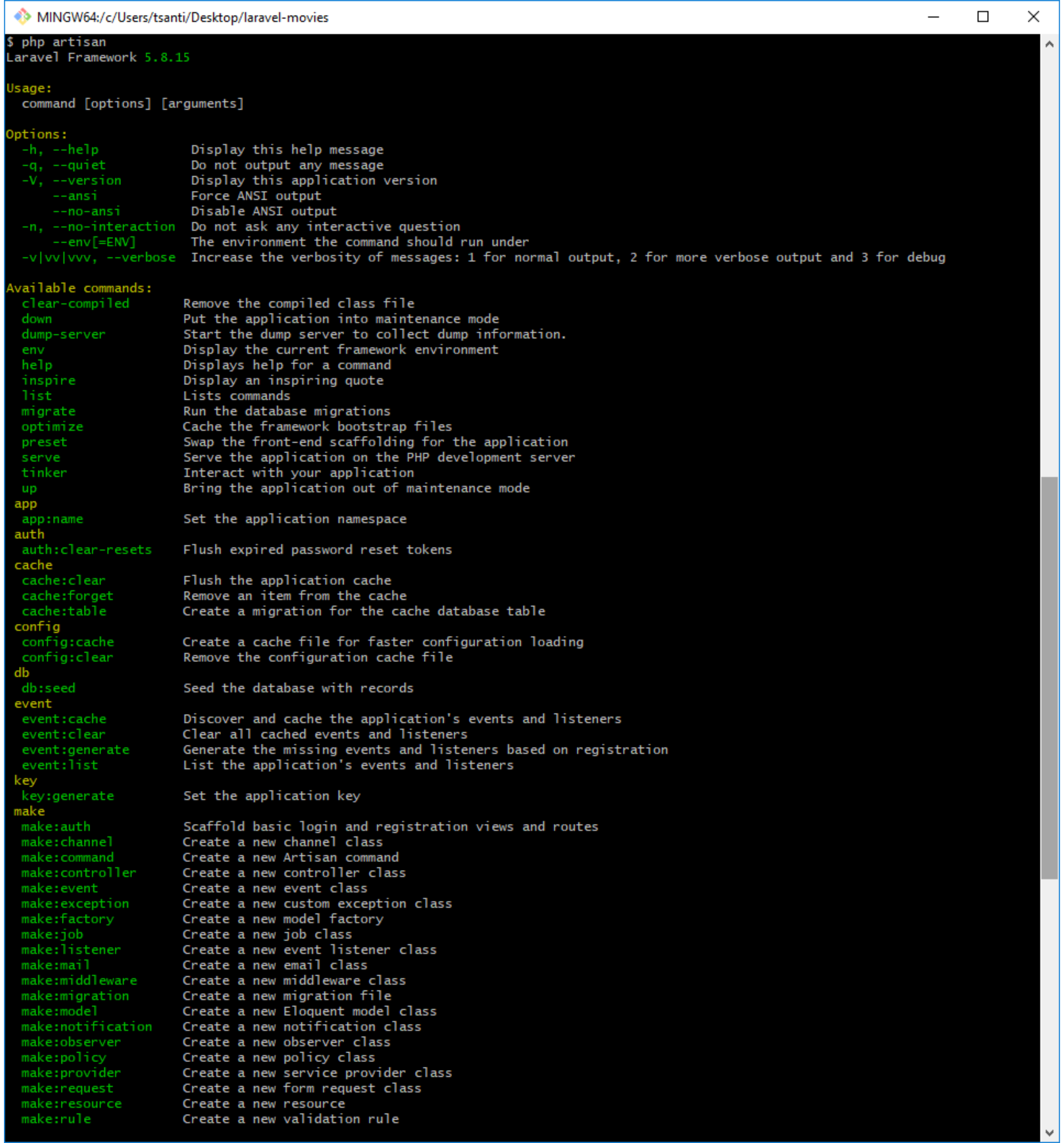
Prologue
Getting Started
Architecture Concepts
The Basics
Frontend
Security
Digging Deeper
Database
Eloquent ORM
Testing
Official Packages



The Basics
Routing
Middleware
CSRF Protection
Controllers
Requests
Responses
Views
Session
Validation
Errors & Logging
Frontend
Blade Templates
Localization
Frontend Scaffolding
Compiling Assets
Eloquent ORM
Getting Started
Relationships
Collections
Mutators
Serialization
Database
Getting Started
Query Builder
Pagination
Migrations
Seeding
Redis

php artisan

Ett hjälpmedel för att snabba upp utvecklingen av webbapplikationer

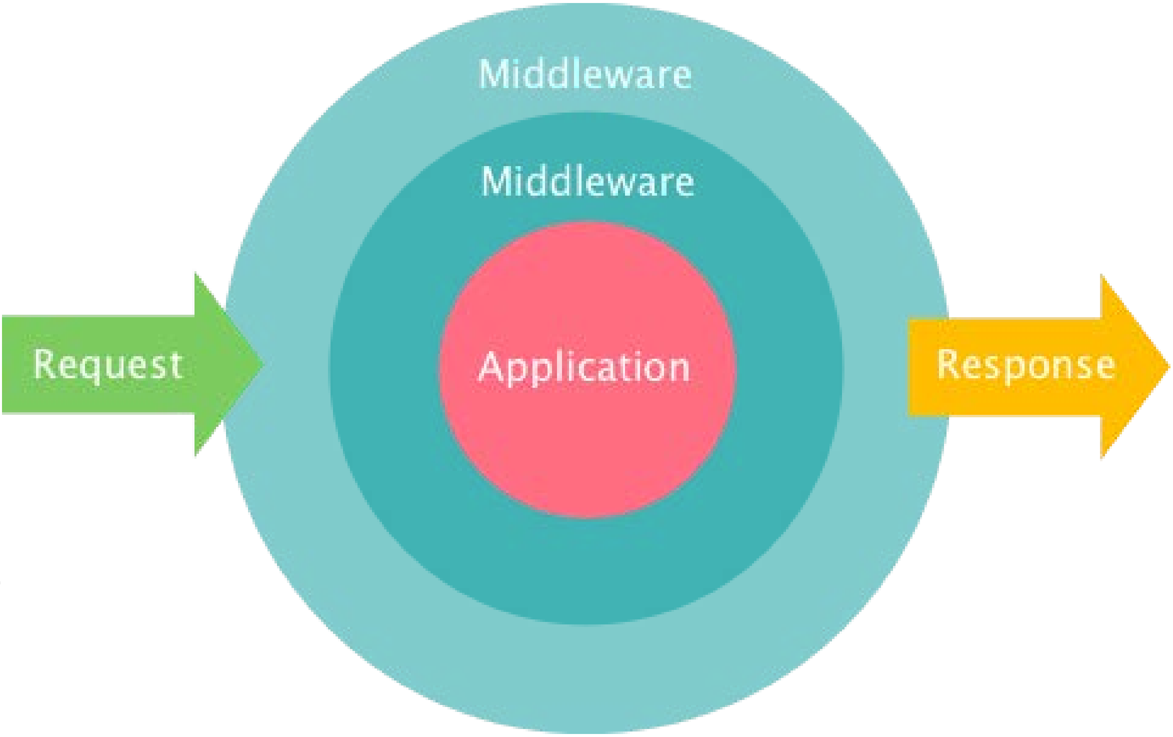


MINGW64:/c/Users/tsanti/Desktop/laravel-movies

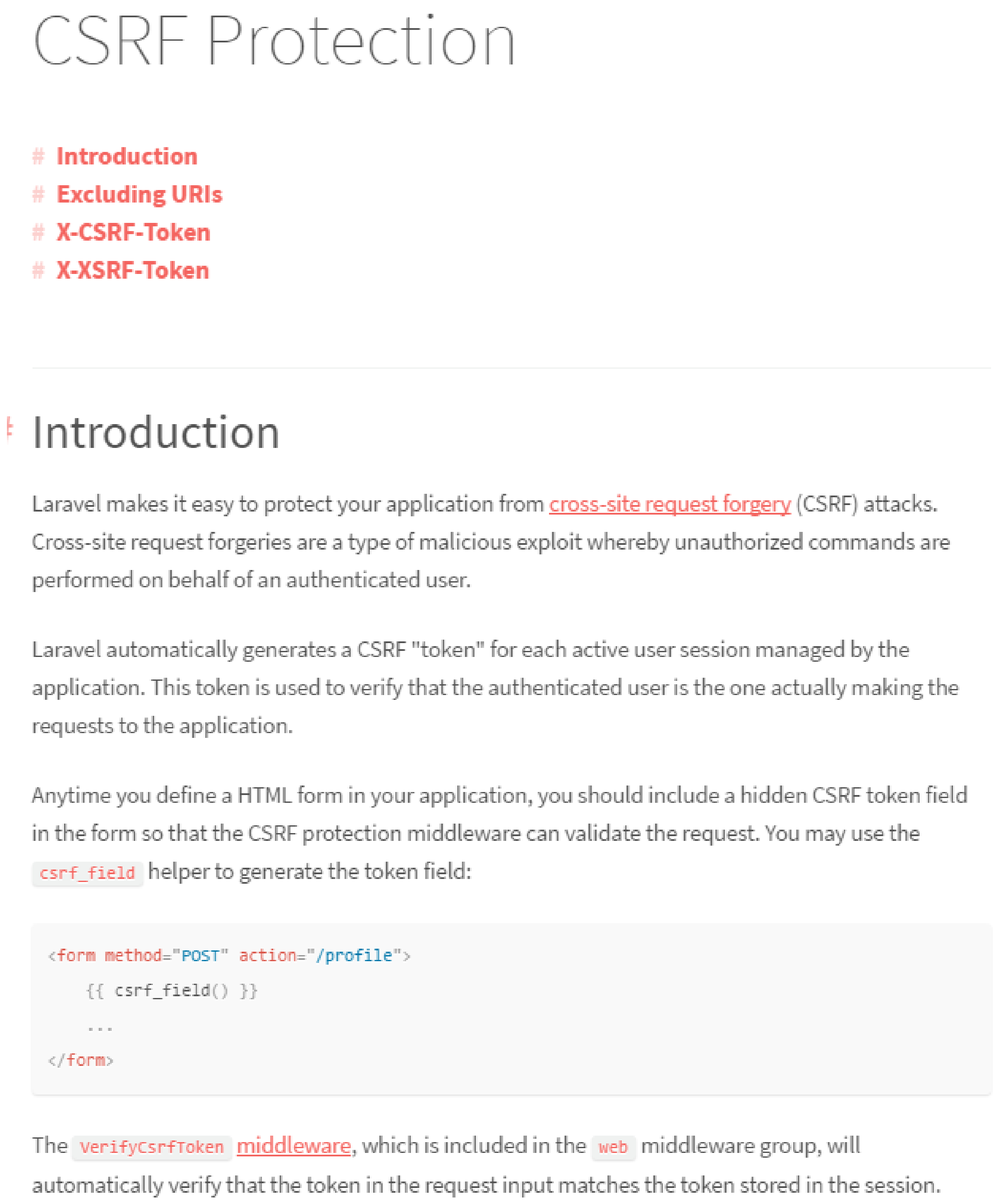
```
$ php artisan
Laravel Framework 5.8.15

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
      --env[=ENV]       The environment the command should run under
  -v|vv|vvv, --verbose  Increase the verbosity of messages: 1 for normal output, 2 for more verbose output and 3 for debug

Available commands:
  clear-compiled       Remove the compiled class file
  down                 Put the application into maintenance mode
  dump-server          Start the dump server to collect dump information.
  env                  Display the current framework environment
  help                 Displays help for a command
  inspire              Display an inspiring quote
  list                 Lists commands
  migrate              Run the database migrations
  optimize             Cache the framework bootstrap files
  preset               Swap the front-end scaffolding for the application
  serve                Serve the application on the PHP development server
  tinker               Interact with your application
  up                   Bring the application out of maintenance mode
 app
  app:name             Set the application namespace
 auth
  auth:clear-resets    Flush expired password reset tokens
 cache
  cache:clear          Flush the application cache
  cache:forget         Remove an item from the cache
  cache:table          Create a migration for the cache database table
 config
  config:cache         Create a cache file for faster configuration loading
  config:clear         Remove the configuration cache file
 db
  db:seed              Seed the database with records
 event
  event:cache          Discover and cache the application's events and listeners
  event:clear          Clear all cached events and listeners
  event:generate       Generate the missing events and listeners based on registration
  event:list           List the application's events and listeners
 key
  key:generate         Set the application key
 make
  make:auth            Scaffold basic login and registration views and routes
  make:channel         Create a new channel class
  make:command         Create a new Artisan command
  make:controller      Create a new controller class
  make:event           Create a new event class
  make:exception       Create a new custom exception class
  make:factory         Create a new model factory
  make:job             Create a new job class
  make:listener        Create a new event listener class
  make:mail            Create a new email class
  make:middleware      Create a new middleware class
  make:migration       Create a new migration file
  make:model           Create a new Eloquent model class
  make:notification    Create a new notification class
  make:observer        Create a new observer class
  make:policy          Create a new policy class
  make:provider        Create a new service provider class
  make:request         Create a new form request class
  make:resource        Create a new resource
  make:rule            Create a new validation rule
```

Middleware

https://laravel.com/docs/5.8/middleware



Middleware
- Vi har vi sett det innan



CSRF Protection

- Introduction
- Excluding URIs
- X-CSRF-Token
- X-XSRF-Token

Introduction

Laravel makes it easy to protect your application from cross-site request forgery (CSRF) attacks. Cross-site request forgeries are a type of malicious exploit whereby unauthorized commands are performed on behalf of an authenticated user.

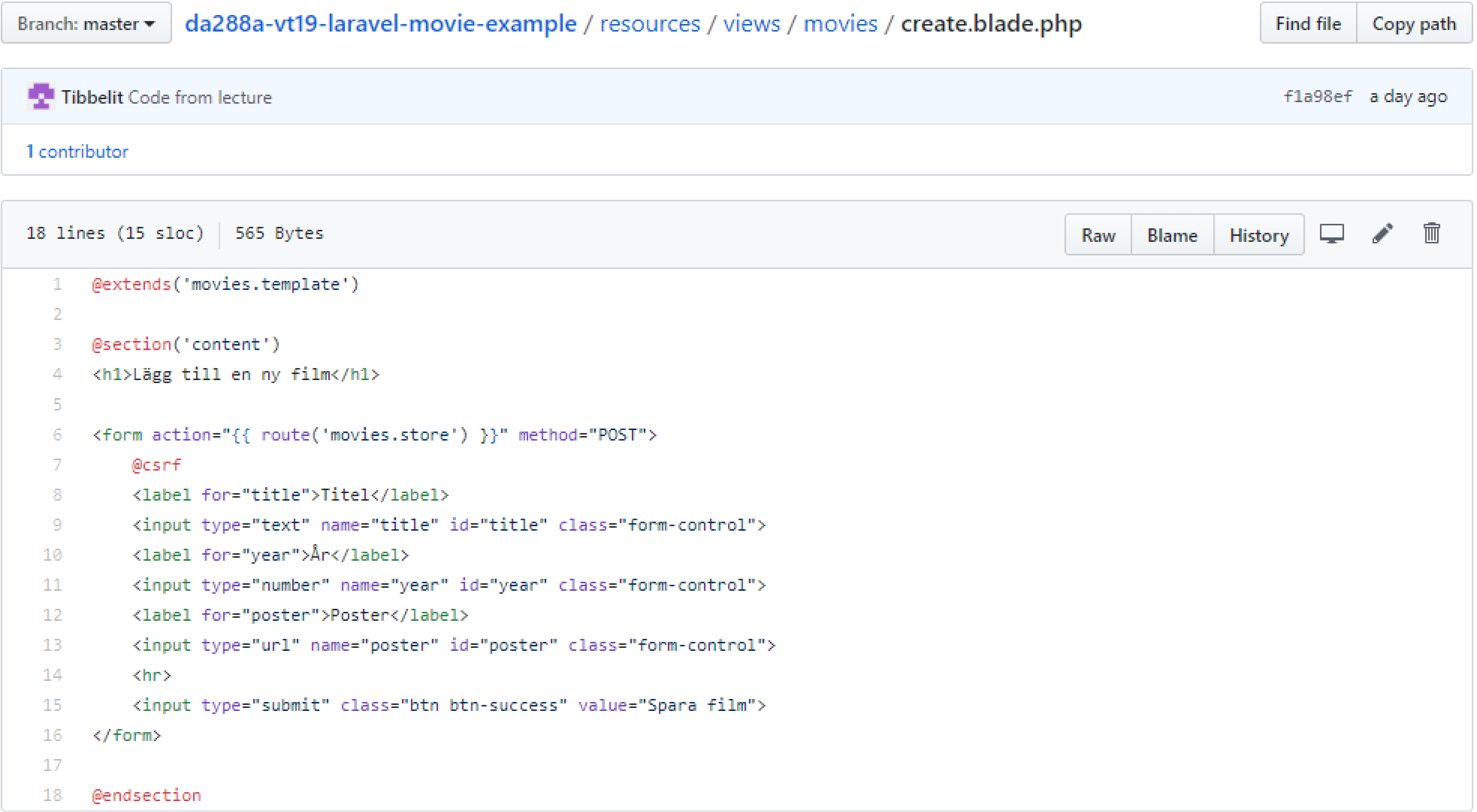
Laravel automatically generates a CSRF "token" for each active user session managed by the application. This token is used to verify that the authenticated user is the one actually making the requests to the application.

Anytime you define a HTML form in your application, you should include a hidden CSRF token field in the form so that the CSRF protection middleware can validate the request. You may use the `csrf_field` helper to generate the token field:

```
<form method="POST" action="/profile">
    {{ csrf_field() }}
    ...
</form>
```

The `VerifyCsrfToken` middleware, which is included in the `web` middleware group, will automatically verify that the token in the request input matches the token stored in the session.

Exempelanvändning av middleware - CSRF



da288a-vt19-laravel-movie-example / resources / views / movies / create.blade.php

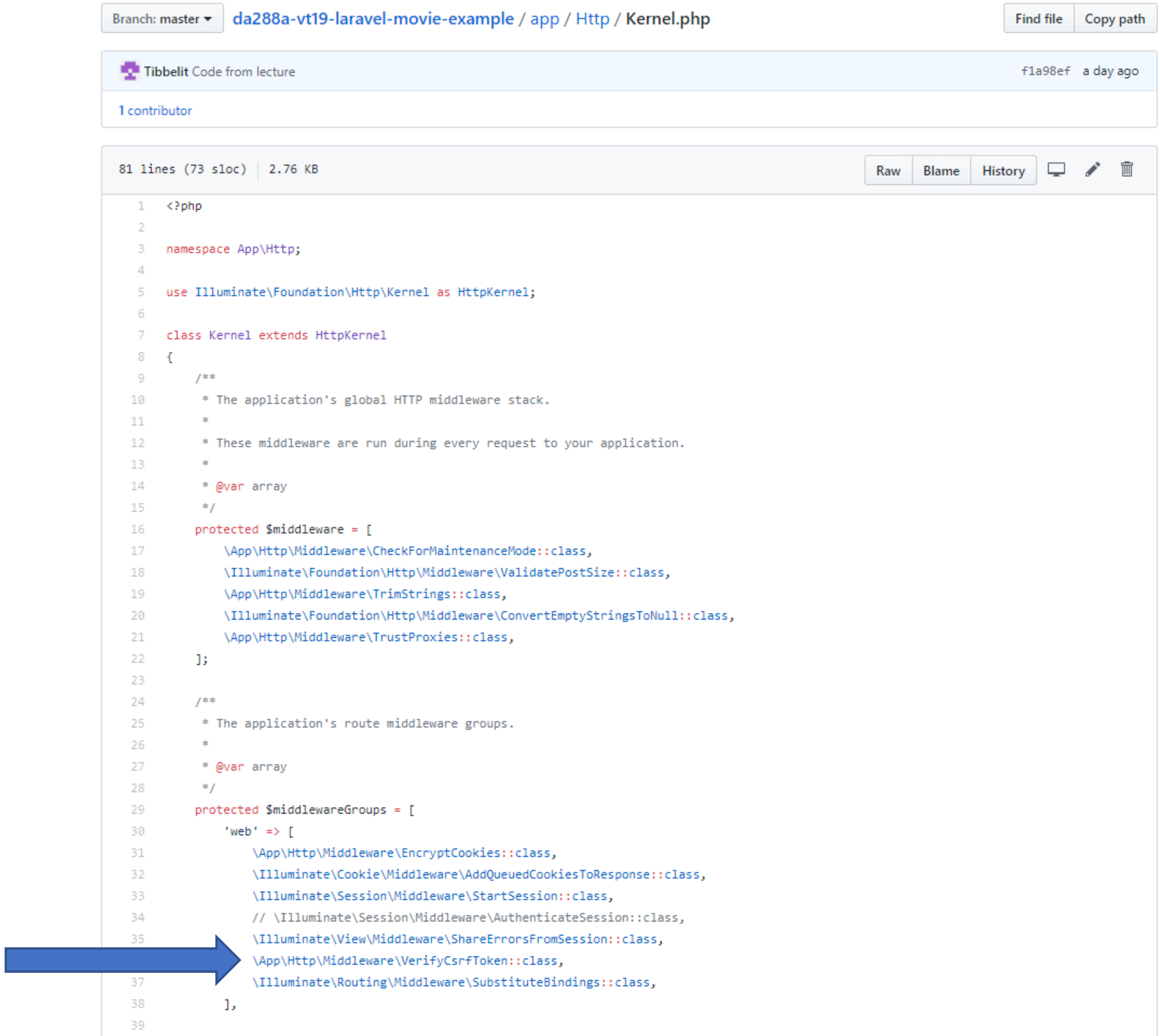
Tibbelit Code from lecture f1a98ef a day ago

1 contributor

18 lines (15 sloc) 565 Bytes

```
@extends('movies.template')

@section('content')
<h1>Lägg till en ny film</h1>

<form action="{{ route('movies.store') }}" method="POST">
    @csrf
    <label for="title">Titel</label>
    <input type="text" name="title" id="title" class="form-control">
    <label for="year">År</label>
    <input type="number" name="year" id="year" class="form-control">
    <label for="poster">Poster</label>
    <input type="url" name="poster" id="poster" class="form-control">
    <hr>
    <input type="submit" class="btn btn-success" value="Spara film">
</form>

@endsection
```

Vilka middleware används, och när?



da288a-vt19-laravel-movie-example / app / Http / Kernel.php

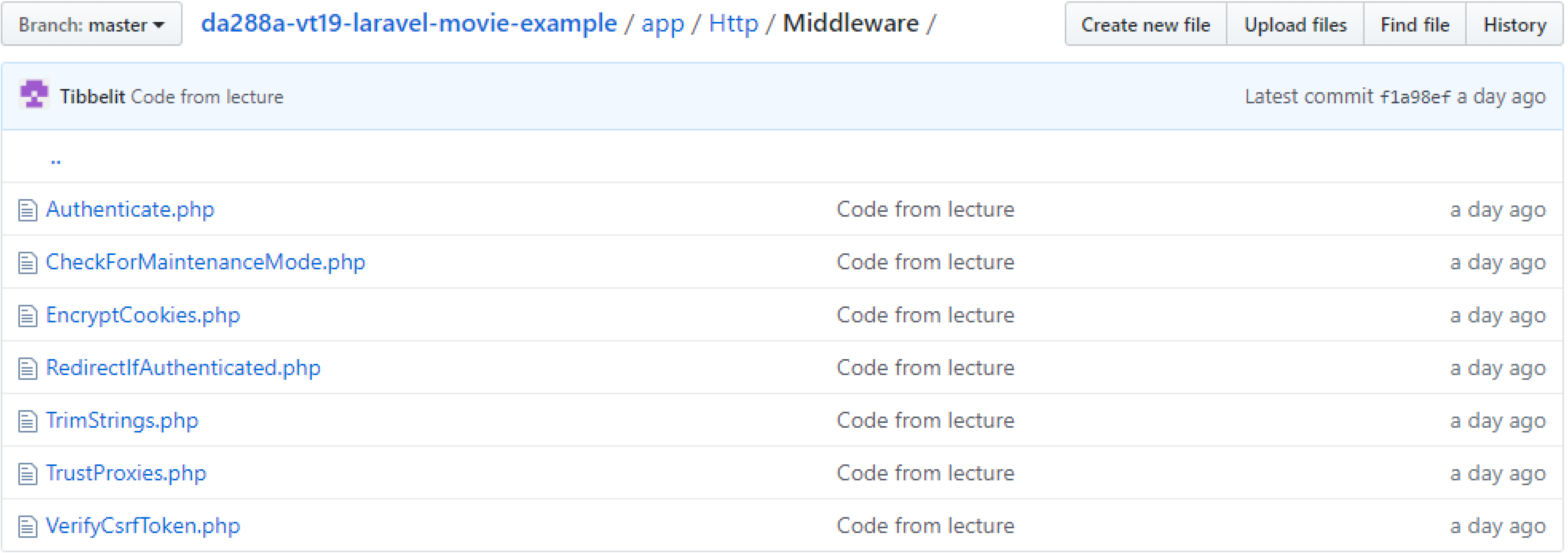
Tibbelit Code from lecture f1a98ef a day ago

1 contributor

81 lines (73 sloc) | 2.76 KB

```php
<?php

namespace App\Http;

use Illuminate\Foundation\Http\Kernel as HttpKernel;

class Kernel extends HttpKernel
{
    /**
     * The application's global HTTP middleware stack.
     *
     * These middleware are run during every request to your application.
     *
     * @var array
     */
    protected $middleware = [
        \App\Http\Middleware\CheckForMaintenanceMode::class,
        \Illuminate\Foundation\Http\Middleware\ValidatePostSize::class,
        \App\Http\Middleware\TrimStrings::class,
        \Illuminate\Foundation\Http\Middleware\ConvertEmptyStringsToNull::class,
        \App\Http\Middleware\TrustProxies::class,
    ];

    /**
     * The application's route middleware groups.
     *
     * @var array
     */
    protected $middlewareGroups = [
        'web' => [
            \App\Http\Middleware\EncryptCookies::class,
            \Illuminate\Cookie\Middleware\AddQueuedCookiesToResponse::class,
            \Illuminate\Session\Middleware\StartSession::class,
            // \Illuminate\Session\Middleware\AuthenticateSession::class,
            \Illuminate\View\Middleware\ShareErrorsFromSession::class,
            \App\Http\Middleware\VerifyCsrfToken::class,
            \Illuminate\Routing\Middleware\SubstituteBindings::class,
        ],

```

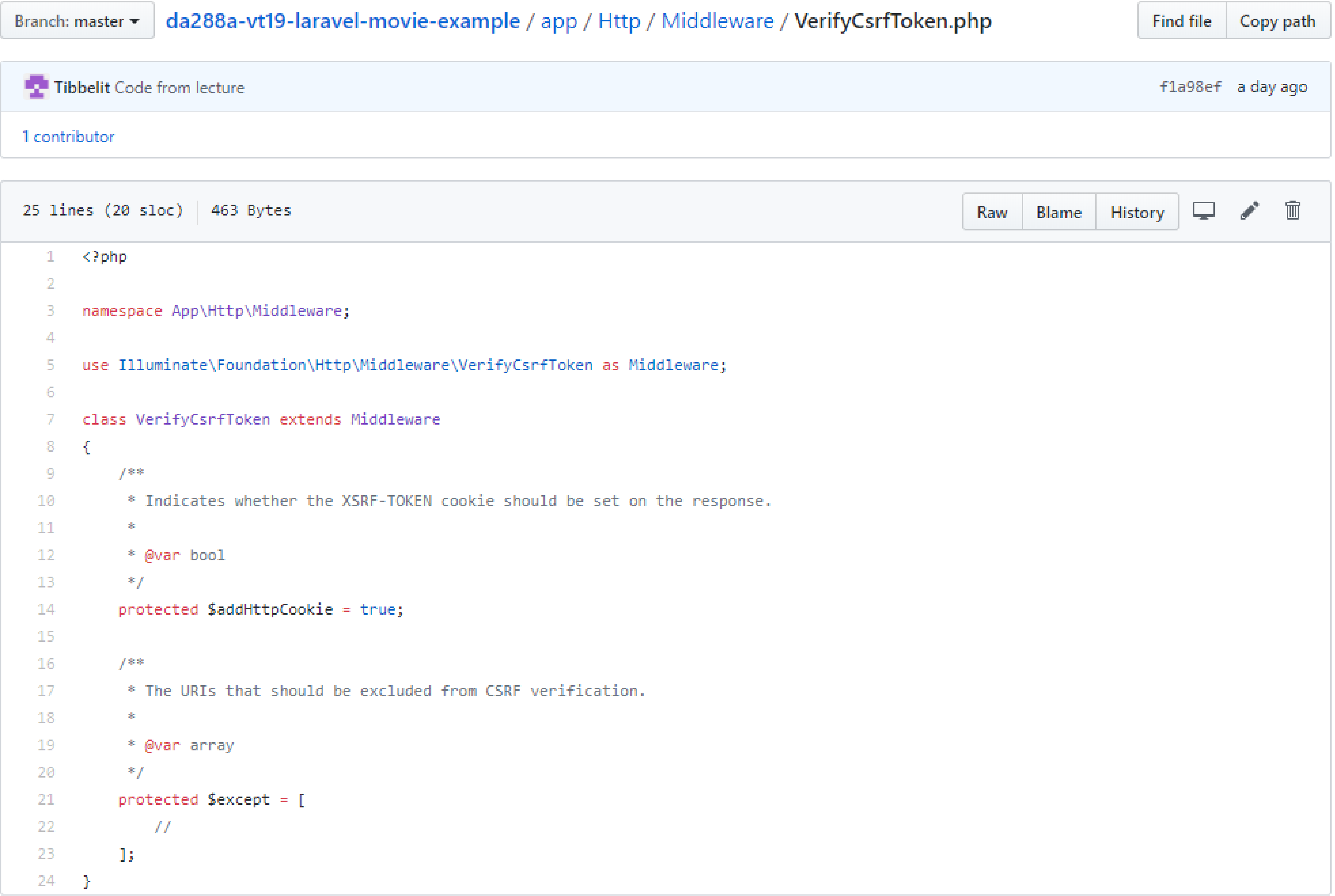
Var finns middleware?



da288a-vt19-laravel-movie-example / app / Http / Middleware /

Tibbelit Code from lecture — Latest commit f1a98ef a day ago

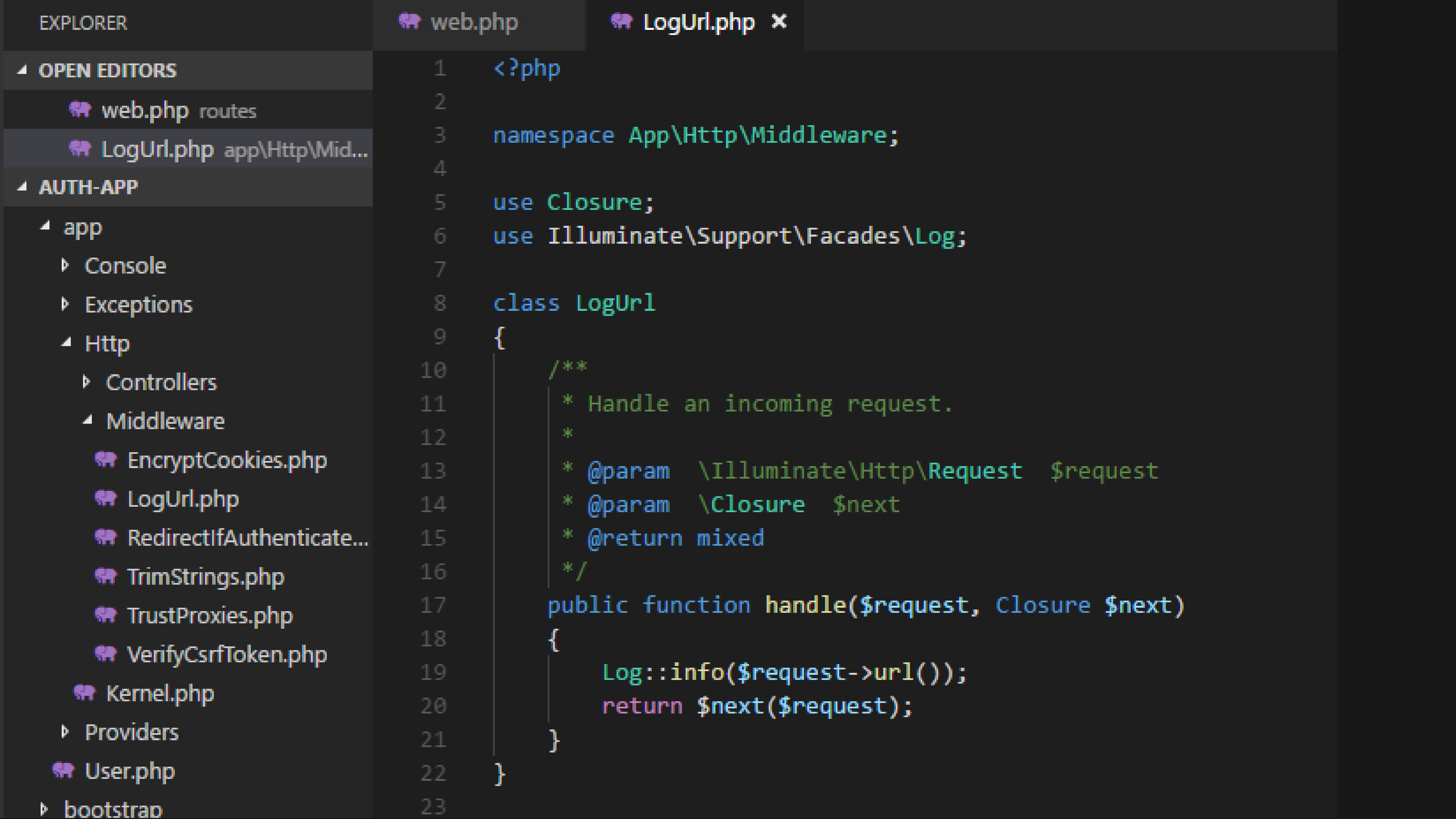
..		
Authenticate.php	Code from lecture	a day ago
CheckForMaintenanceMode.php	Code from lecture	a day ago
EncryptCookies.php	Code from lecture	a day ago
RedirectIfAuthenticated.php	Code from lecture	a day ago
TrimStrings.php	Code from lecture	a day ago
TrustProxies.php	Code from lecture	a day ago
VerifyCsrfToken.php	Code from lecture	a day ago


Branch: master ▾
da288a-vt19-laravel-movie-example / app / Http / Middleware / VerifyCsrfToken.php
Find file
Copy path
Tibbelit Code from lecture
f1a98ef a day ago
1 contributor
25 lines (20 sloc) 463 Bytes
Raw
Blame
History

```php
<?php

namespace App\Http\Middleware;

use Illuminate\Foundation\Http\Middleware\VerifyCsrfToken as Middleware;

class VerifyCsrfToken extends Middleware
{
    /**
     * Indicates whether the XSRF-TOKEN cookie should be set on the response.
     *
     * @var bool
     */
    protected $addHttpCookie = true;

    /**
     * The URIs that should be excluded from CSRF verification.
     *
     * @var array
     */
    protected $except = [
        //
    ];
}
```

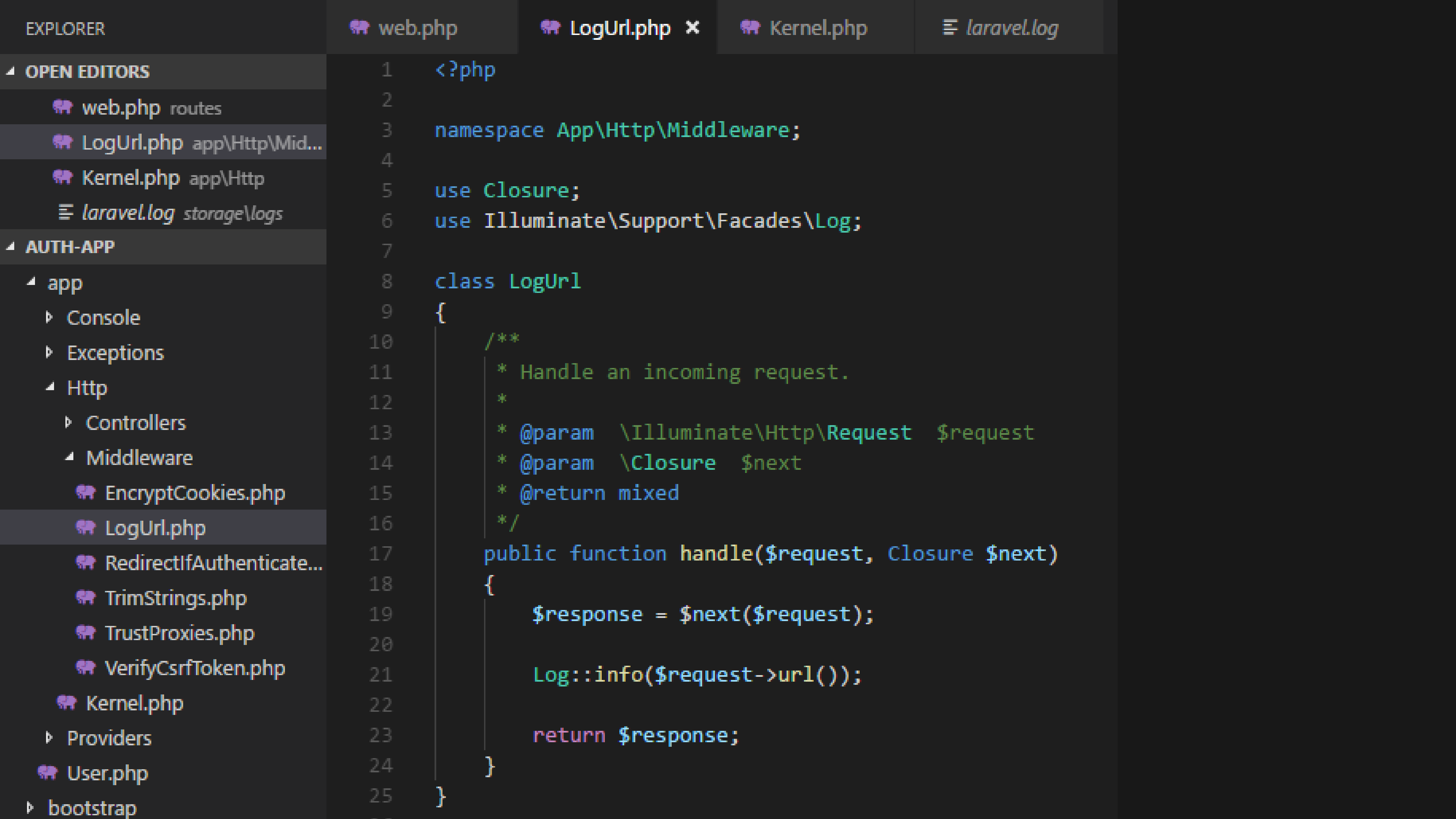
php artisan make:middleware LogUrl

Skapa en middleware för att logga anropad URL



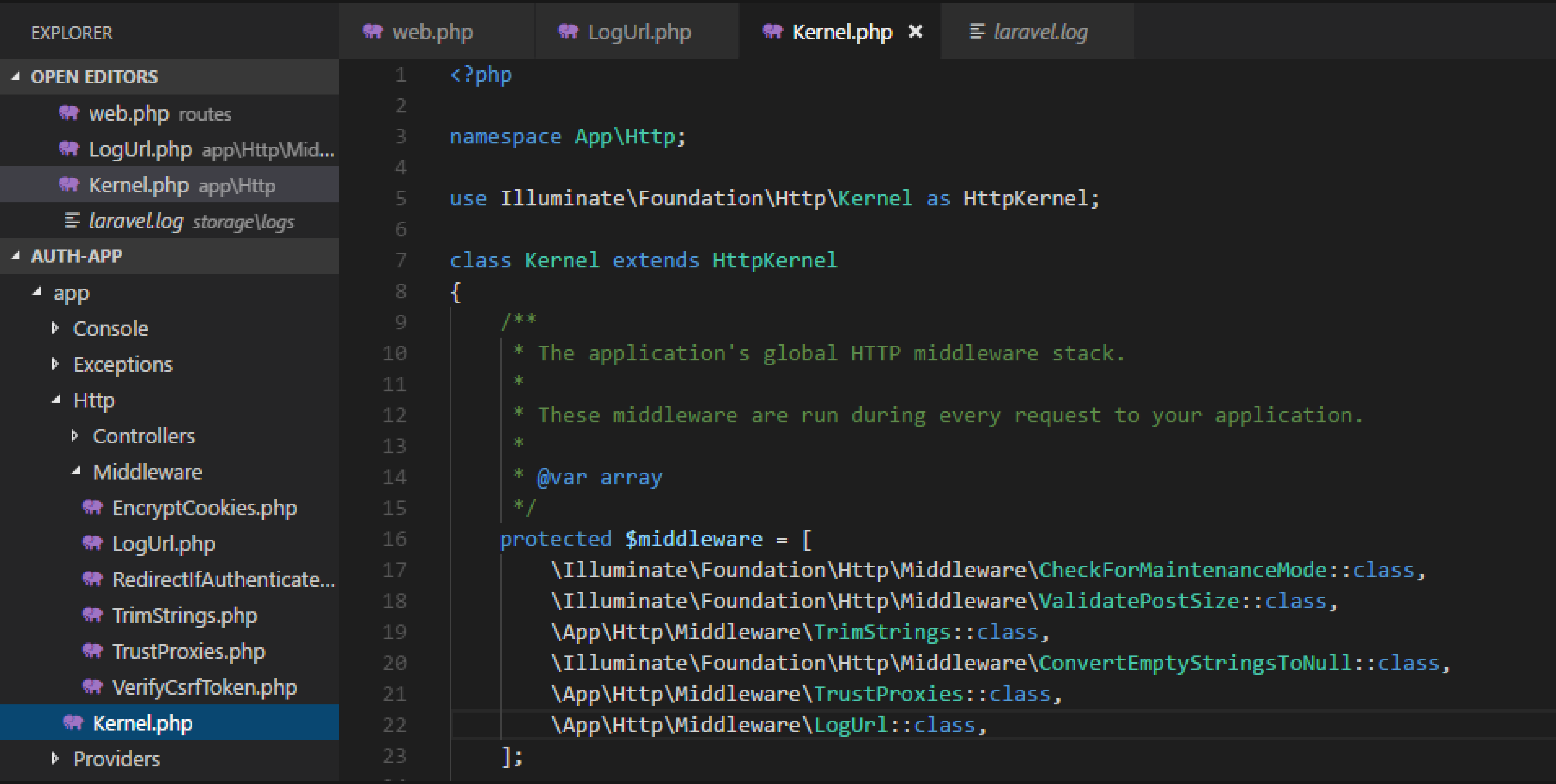
```php
<?php

namespace App\Http\Middleware;

use Closure;
use Illuminate\Support\Facades\Log;

class LogUrl
{
    /**
     * Handle an incoming request.
     *
     * @param  \Illuminate\Http\Request  $request
     * @param  \Closure  $next
     * @return mixed
     */
    public function handle($request, Closure $next)
    {
        Log::info($request->url());
        return $next($request);
    }
}

```

Efter ett request?



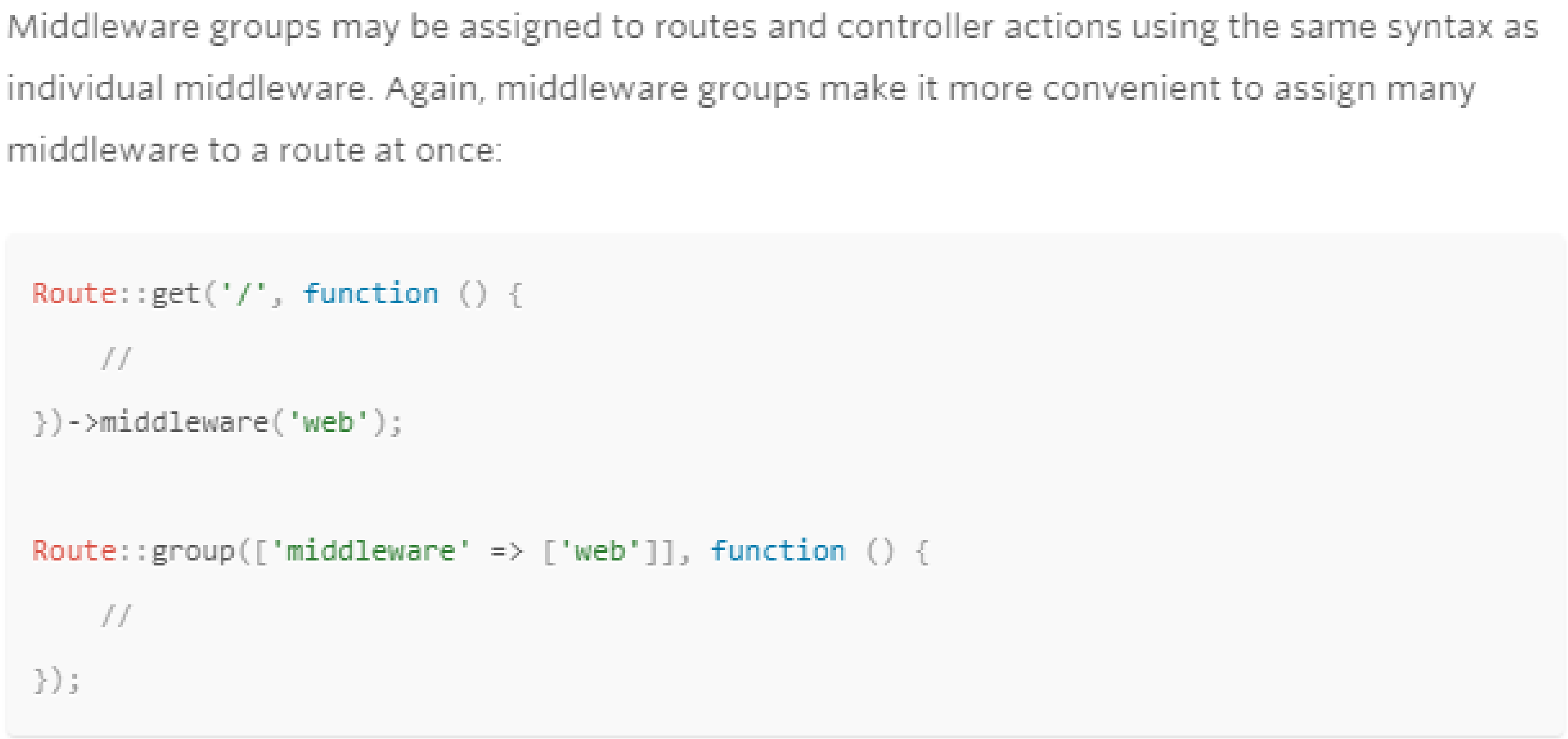
```php
<?php

namespace App\Http\Middleware;

use Closure;
use Illuminate\Support\Facades\Log;

class LogUrl
{
    /**
     * Handle an incoming request.
     *
     * @param  \Illuminate\Http\Request  $request
     * @param  \Closure  $next
     * @return mixed
     */
    public function handle($request, Closure $next)
    {
        $response = $next($request);

        Log::info($request->url());

        return $response;
    }
}
```

Att registrera en middleware



```php
<?php

namespace App\Http;

use Illuminate\Foundation\Http\Kernel as HttpKernel;

class Kernel extends HttpKernel
{
    /**
     * The application's global HTTP middleware stack.
     *
     * These middleware are run during every request to your application.
     *
     * @var array
     */
    protected $middleware = [
        \Illuminate\Foundation\Http\Middleware\CheckForMaintenanceMode::class,
        \Illuminate\Foundation\Http\Middleware\ValidatePostSize::class,
        \App\Http\Middleware\TrimStrings::class,
        \Illuminate\Foundation\Http\Middleware\ConvertEmptyStringsToNull::class,
        \App\Http\Middleware\TrustProxies::class,
        \App\Http\Middleware\LogUrl::class,
    ];
```

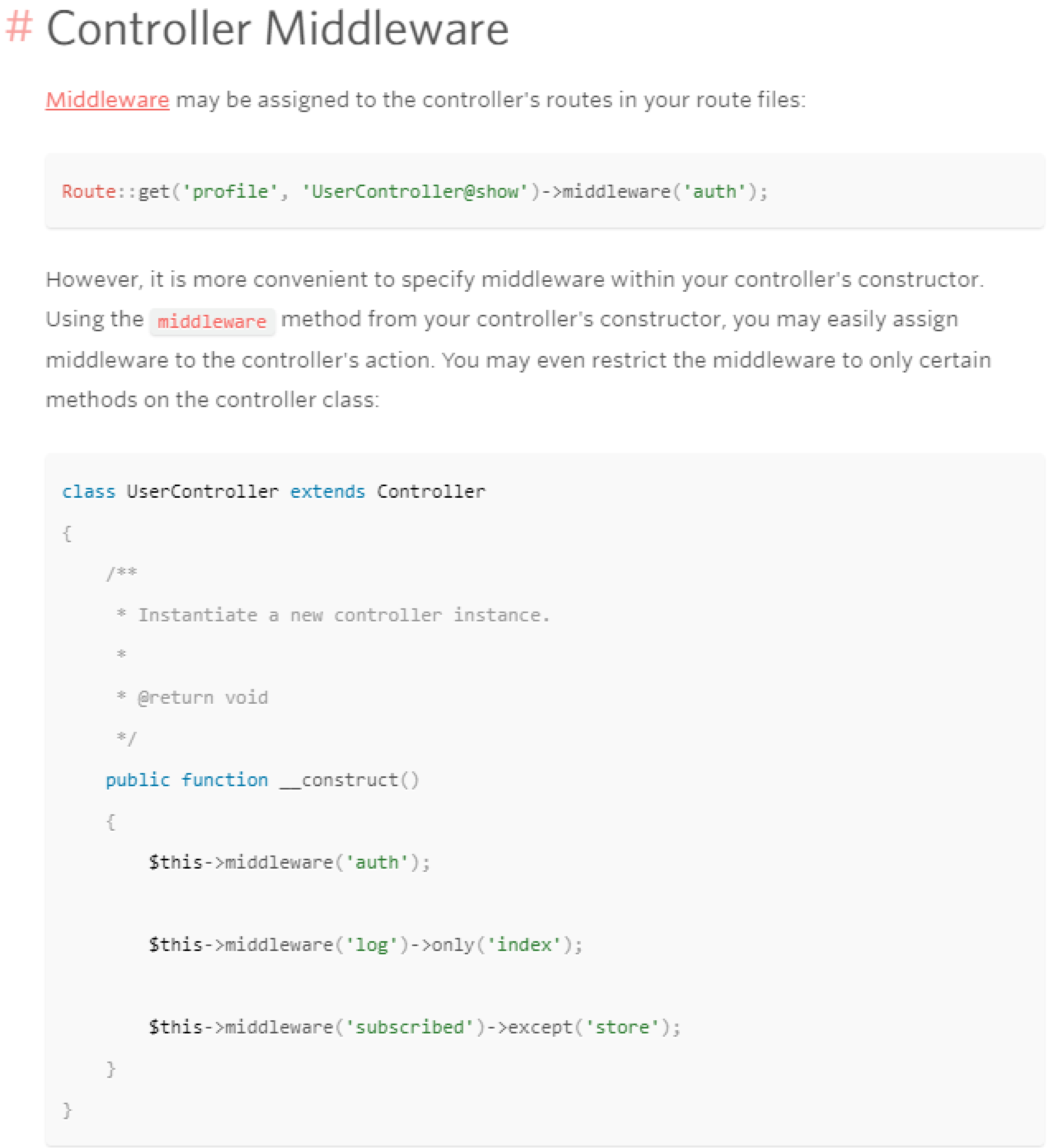
Middleware som ej gäller alla routes



Middleware groups may be assigned to routes and controller actions using the same syntax as individual middleware. Again, middleware groups make it more convenient to assign many middleware to a route at once:

```
Route::get('/', function () {
    //
})->middleware('web');

Route::group(['middleware' => ['web']], function () {
    //
});
```

https://laravel.com/docs/5.8/middleware



Controller Middleware

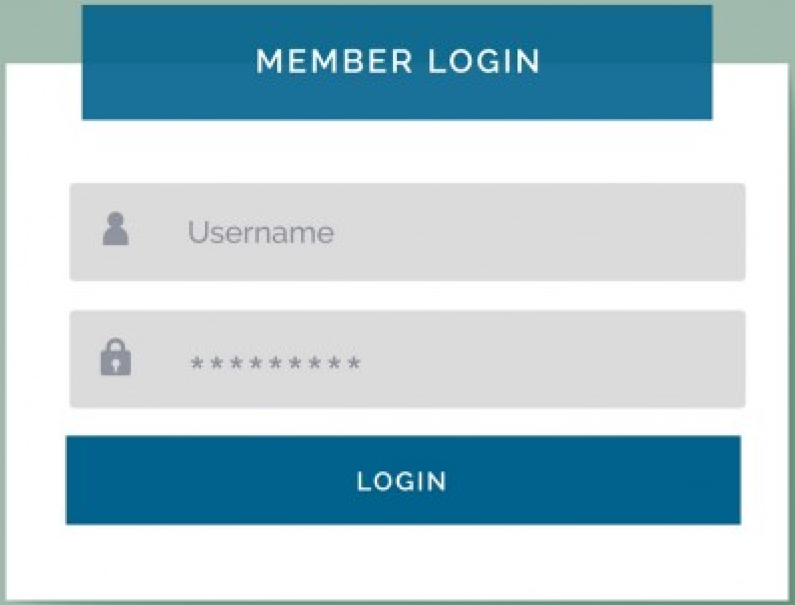
Middleware may be assigned to the controller's routes in your route files:

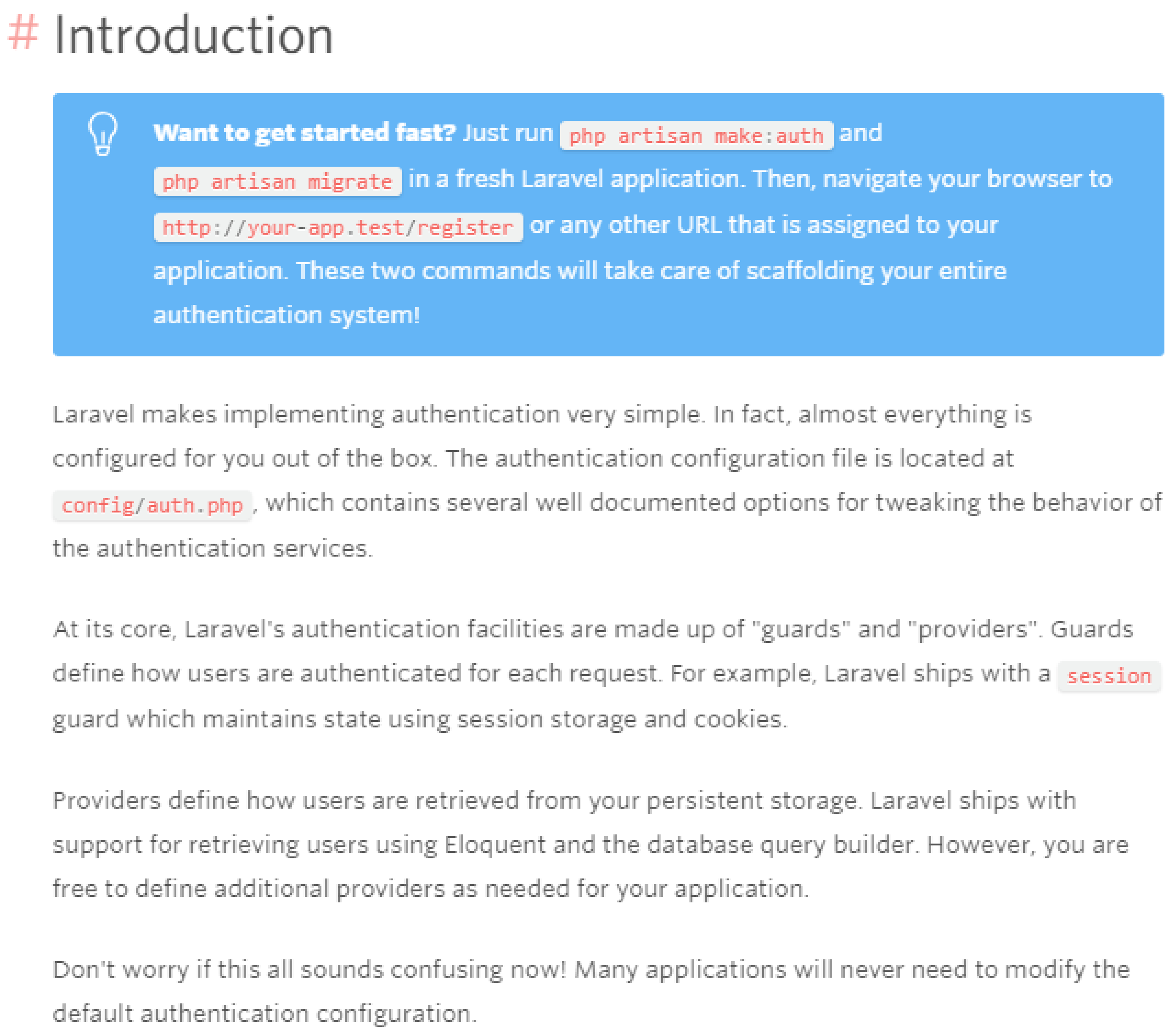
```
Route::get('profile', 'UserController@show')->middleware('auth');
```

However, it is more convenient to specify middleware within your controller's constructor. Using the `middleware` method from your controller's constructor, you may easily assign middleware to the controller's action. You may even restrict the middleware to only certain methods on the controller class:

```
class UserController extends Controller
{
    /**
     * Instantiate a new controller instance.
     *
     * @return void
     */
    public function __construct()
    {
        $this->middleware('auth');

        $this->middleware('log')->only('index');

        $this->middleware('subscribed')->except('store');
    }
}
```

Demo - middleware


MEMBER LOGIN
Username
****
LOGIN



Introduction

Want to get started fast? Just run `php artisan make:auth` and `php artisan migrate` in a fresh Laravel application. Then, navigate your browser to `http://your-app.test/register` or any other URL that is assigned to your application. These two commands will take care of scaffolding your entire authentication system!

Laravel makes implementing authentication very simple. In fact, almost everything is configured for you out of the box. The authentication configuration file is located at `config/auth.php`, which contains several well documented options for tweaking the behavior of the authentication services.

At its core, Laravel's authentication facilities are made up of "guards" and "providers". Guards define how users are authenticated for each request. For example, Laravel ships with a `session` guard which maintains state using session storage and cookies.

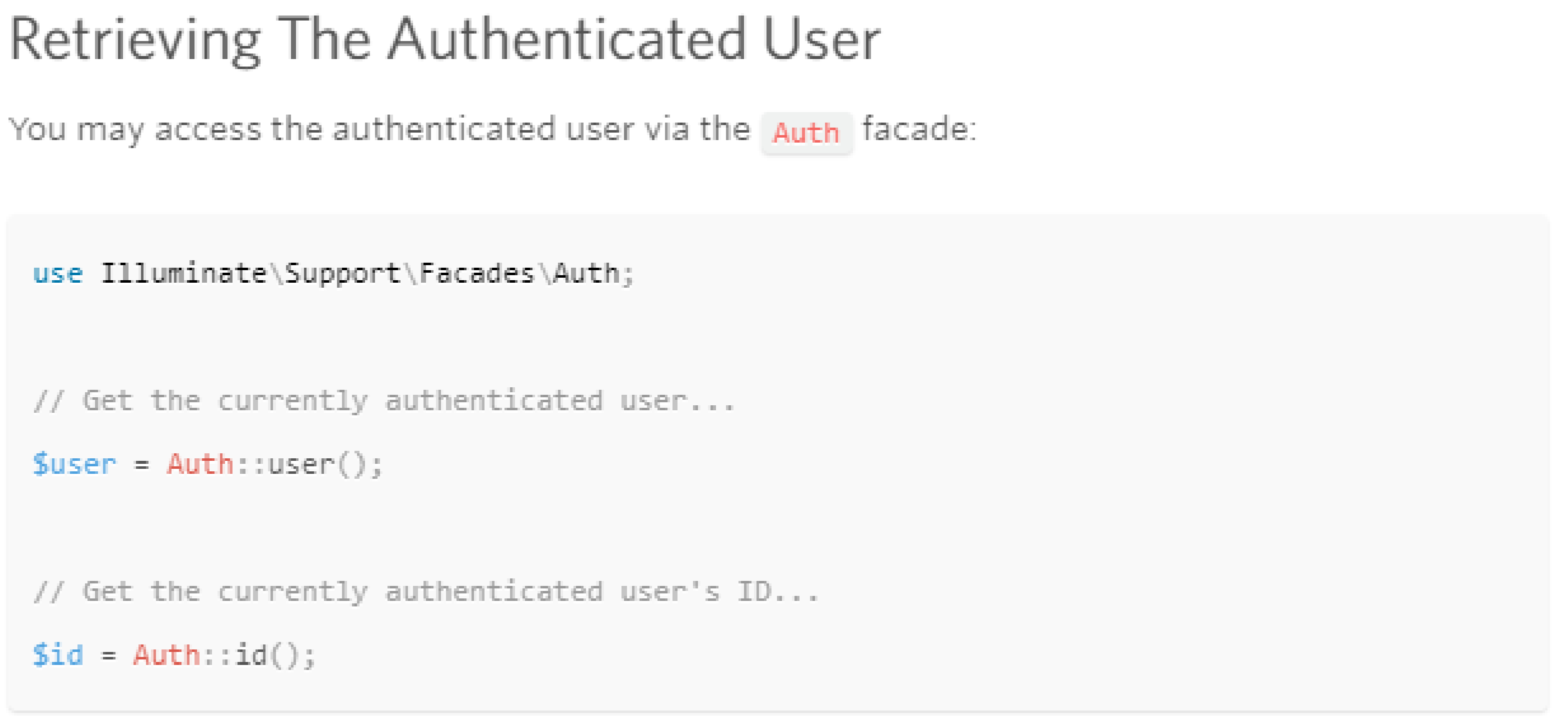
Providers define how users are retrieved from your persistent storage. Laravel ships with support for retrieving users using Eloquent and the database query builder. However, you are free to define additional providers as needed for your application.

Don't worry if this all sounds confusing now! Many applications will never need to modify the default authentication configuration.

php artisan make:auth

- Skapar routes för inloggning
  - Kontrollera routes genom kommando: *php artisan route:list* i terminalen
- Skapar Controller för *Auth* (mappen *Auth* under *Controllers*)
- Skapar vyer för inloggning

- Vi behöver
  - Skapa en databas
  - Uppdatera vår .env-fil så att den stämmer överns med databasen
  - Migrera databas

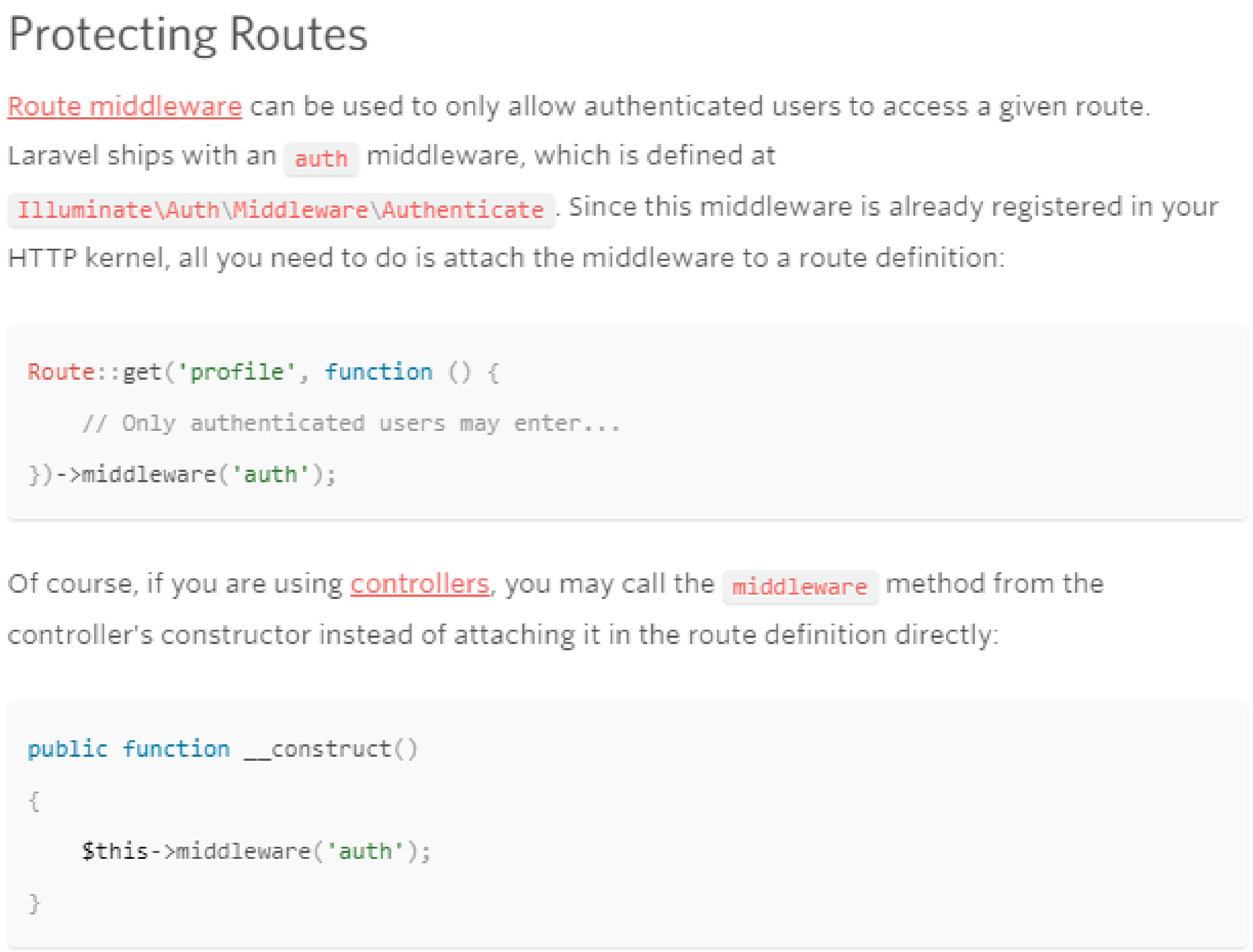


Retrieving The Authenticated User

You may access the authenticated user via the `Auth` facade:

```
use Illuminate\Support\Facades\Auth;

// Get the currently authenticated user...
$user = Auth::user();

// Get the currently authenticated user's ID...
$id = Auth::id();
```

https://laravel.com/docs/5.8/authentication



Protecting Routes

Route middleware can be used to only allow authenticated users to access a given route. Laravel ships with an `auth` middleware, which is defined at `Illuminate\Auth\Middleware\Authenticate`. Since this middleware is already registered in your HTTP kernel, all you need to do is attach the middleware to a route definition:

```
Route::get('profile', function () {
    // Only authenticated users may enter...
})->middleware('auth');
```

Of course, if you are using controllers, you may call the `middleware` method from the controller's constructor instead of attaching it in the route definition directly:

```
public function __construct()
{
    $this->middleware('auth');
}
```

https://laravel.com/docs/5.8/authentication

Demo!

Sammanfattning av denna del i kursen

Reflektion i kursen

- Beskriv följande delar i kursen:
  - PHP
  - Pakethanterare
  - Versionshantering
  - Lumen
  - Laravel
- Inom MVC och webbramverk beskriv följande delar i kursen:
  - Routes
  - Controllers
  - Models
  - Views
  - ORM
  - Migration & seeds
  - Authentisering
  - Middleware

Kommande delar i kursen

- Testning
- Refaktorisering
- Integration
- Deployment
- Delivery
- => Att publicera ett projekt