Howdy!

Writing quality PHP code with Laravel

Joey Kudish

jkudish.com

slides.jkudish.com

Twitter: @jkudish

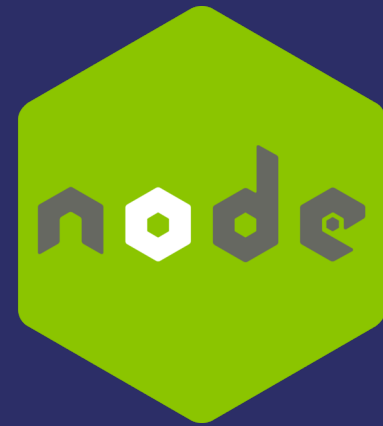# My Journey

https://sparkdev.io

https://theimagesalon.com

+

RethinkDB

(+ React)
(+ Experimentation with Go and others)

Full circle back
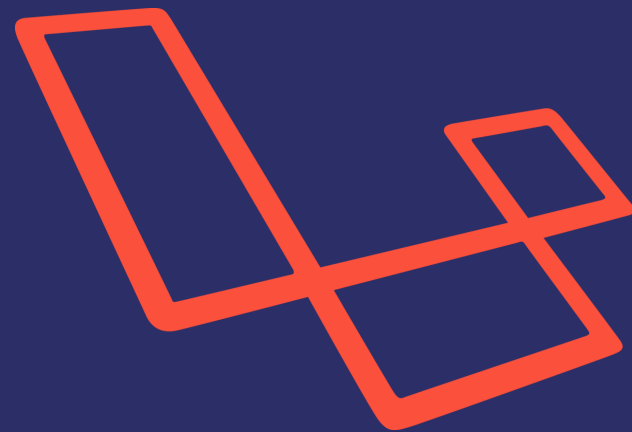
So what is Laravel?

MVC Framework written in PHP - MIT License

Started in 2011 by Taylor Otwell

Now: Version 5.5 (uses PHP 7 features and requires it)

Uses some Symfony components under the hood when it makes sense to

At its core it's a fairly simple and magical Dependency Injection/Service container with even more magic sprinkled on top of it.

Laravel's biggest focus is on the developer experience.

```
$ composer global require "laravel/installer"
$ laravel new confoo-demo
```

Laravel is optimized for being run according to its conventions. Avoid doing things differently unless you have a good reason to.

```php
<?php

namespace TIS;

use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\SoftDeletes;

class Gallery extends Model
{
    use SoftDeletes;
```

Almost everything is configurable or overwrite-able if need be though!

config
  xero
    config.php
  app.php
  auth.php
  blade-inline-svg.php
  blade-javascript.php
  broadcasting.php
  bugsnag.php
  cache.php
  compile.php
  database.php
  eyewitness.php
  failed-job-monitor.php
  filesystems.php
  horizon.php
  invoices.php
  jwt.php
  mail.php
  newsletter.php
  queue.php
  scout.php
  services.php
  session.php
  tis.php
  verifi.php
  view.php

```php
<?php
return [

    /*
    |--------------------------------------------------------------------------
    | Application Name
    |--------------------------------------------------------------------------
    |
    | This value is the name of your application. This value is used when the
    | framework needs to place the application's name in a notification or
    | any other location as required by the application or its packages.
    */

    'name' => 'The Image Salon',

    /*
    |--------------------------------------------------------------------------
    | Application Environment
    |--------------------------------------------------------------------------
    |
    | This value determines the "environment" your application is currently
    | running in. This may determine how you prefer to configure various
    | services your application utilizes. Set this in your ".env" file.
    |
    */

    'env' => env('APP_ENV', 'local'),
```

```
config('name');
```

```
$ php artisan make:model Presentation -m -c -f -r
```

```
app/Presentation.php
app/Http/Controllers/PresentationController.php
database/migrations/2017_12_03_002326_create_presentations_table.php
database/factories/PresentationFactory.php
```

```php
<?php

# database/migrations/2017_12_03_002326_create_presentations_table.php

use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;

class CreatePresentationsTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('presentations', function (Blueprint $table) {
            $table->increments('id');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('presentations');
    }
}
```

```php
<?php

use Faker\Generator as Faker;

$factory->define(TIS\Presentation::class, function (Faker $faker) {
    return [
        //
    ];
});
```

```php
<?php

namespace TIS;

use Illuminate\Database\Eloquent\Model;

class Presentation extends Model
{
    //
}
```

```php
<?php

namespace TIS\Http\Controllers;

use TIS\Presentation;
use Illuminate\Http\Request;

class PresentationController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        //
    }

    /**
     * Show the form for creating a new resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function create()
    {
        //
    }

    /**
     * Store a newly created resource in storage.
     *
     * @param  \Illuminate\Http\Request  $request
     * @return \Illuminate\Http\Response
     */
    public function store(Request $request)
    {
        //
    }

    /**
     * Display the specified resource.
     *
     * @param  \TIS\Presentation  $presentation
     * @return \Illuminate\Http\Response
     */
    public function show(Presentation $presentation)
    {
        //
    }

    /**
     * Show the form for editing the specified resource.
     *
     * @param  \TIS\Presentation  $presentation
     * @return \Illuminate\Http\Response
     */
    public function edit(Presentation $presentation)
    {
        //
    }

    /**
     * Update the specified resource in storage.
     *
     * @param  \Illuminate\Http\Request  $request
     * @param  \TIS\Presentation  $presentation
     * @return \Illuminate\Http\Response
     */
    public function update(Request $request, Presentation $presentation)
    {
        //
    }

    /**
     * Remove the specified resource from storage.
     *
     * @param  \TIS\Presentation  $presentation
     * @return \Illuminate\Http\Response
     */
    public function destroy(Presentation $presentation)
    {
        //
    }
}
```

```php
<?php

// ...



    /**
     * Store a newly created resource in storage.
     *
     * @param  \Illuminate\Http\Request  $request
     * @return \Illuminate\Http\Response
     */
    public function store(Request $request)
    {
        //
    }

    /**
     * Display the specified resource.
     *
     * @param  \TIS\Presentation  $presentation
     * @return \Illuminate\Http\Response
     */
    public function show(Presentation $presentation)
    {
        //
    }

// ...
```

Request Cycle

```
public/index.php -> autoload.php + service container + http
kernel -> service providers -> routes -> controller -> whatever
```

Service Providers

Providers
- AppServiceProvider.php
- AuthServiceProvider.php
- BroadcastServiceProvider.php
- BusinessHoursServiceProvider.php
- DuskServiceProvider.php
- EventServiceProvider.php
- RouteServiceProvider.php
- SlackServiceProvider.php
- TlSVerifiServiceProvider.php

```php
<?php

namespace TIS\Providers;

use CL\Slack\Payload\ChatPostMessagePayload;
use CL\Slack\Transport\ApiClient;
use Illuminate\Support\ServiceProvider;

class SlackServiceProvider extends ServiceProvider
{
    public function register()
    {
        $this->app->bind(ApiClient::class, function ($token = null) {
            $token = $token ?: config('services.slack.oauth_token');
            return new ApiClient($token);
        });

        $this->app->bind(ChatPostMessagePayload::class, function () {
            return new ChatPostMessagePayload();
        });
    }

    public function boot()
    {
        //
    }
}
```

Kernels

```php
<?php

namespace TIS\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,
        \TIS\Http\Middleware\TrimStrings::class,
        \Illuminate\Foundation\Http\Middleware\ConvertEmptyStringsToNull::class,
        \TIS\Http\Middleware\TrustProxies::class,
    ];

    /**
     * The application's route middleware groups.
     *
     * @var array
     */
    protected $middlewareGroups = [
        'web' => [ // need for dusk to function properly, should mimick clientdash
            \TIS\Http\Middleware\EncryptCookies::class,
            \Illuminate\Cookie\Middleware\AddQueuedCookiesToResponse::class,
            \Illuminate\Session\Middleware\StartSession::class,
            \Illuminate\View\Middleware\ShareErrorsFromSession::class,
            \TIS\Http\Middleware\VerifyCsrfToken::class,
            \Illuminate\Routing\Middleware\SubstituteBindings::class,
            \Laravel\Passport\Http\Middleware\CreateFreshApiToken::class,
            \TIS\Http\Middleware\SetReferralCodeCookie::class,
        ],
        'api' => [
//            'throttle:60,1',
            'bindings',
        ],
    ];

    /**
     * The application's route middleware.
     *
     * These middleware may be assigned to groups or used individually.
     *
     * @var array
     */
    protected $routeMiddleware = [
        'auth' => \Illuminate\Auth\Middleware\Authenticate::class,
        'auth.basic' => \Illuminate\Auth\Middleware\AuthenticateWithBasicAuth::class,
        'bindings' => \Illuminate\Routing\Middleware\SubstituteBindings::class,
        'can' => \Illuminate\Auth\Middleware\Authorize::class,
        'guest' => \TIS\Http\Middleware\RedirectIfAuthenticated::class,
        'mailgun' => \TIS\Http\Middleware\Webhooks\VerifyMailgunInput::class,
        'mailchimp' => \TIS\Http\Middleware\Webhooks\VerifyMailchimpSignature::class,
        'throttle' => \Illuminate\Routing\Middleware\ThrottleRequests::class,
        'checkbrowser' => \TIS\Http\Middleware\CheckBrowser::class,
    ];
}
```

```php
<?php

namespace TIS\Console;

// ...

class Kernel extends ConsoleKernel
{
    /**
     * The Artisan commands provided by your application.
     *
     * @var array
     */
    protected $commands = [
        GetDailyStats::class,
        // ...
    ];

    /**
     * Define the application's command schedule.
     *
     * @param  \Illuminate\Console\Scheduling\Schedule  $schedule
     * @return void
     */
    protected function schedule(Schedule $schedule)
    {
        $schedule->command('horizon:snapshot')->everyFiveMinutes();
        $schedule->job(DailyReport::class)->dailyAt('00:30')->timezone(config('app.timezone'));
    }

    /**
     * Register the Closure based commands for the application.
     *
     * @return void
     */
    protected function commands()
    {
        require base_path('routes/console.php');
    }
}
```

Middleware

```php
<?php

namespace TIS\Http\Middleware;

use Closure;
use Illuminate\Support\Facades\Request;
use Jenssegers\Agent\Agent;

class CheckBrowser
{
    /**
     * Handle an incoming request.
     *
     * @param  \Illuminate\Http\Request $request
     * @param  \Closure $next
     * @return mixed
     */
    public function handle($request, Closure $next)
    {
        $agent = new Agent();
        $browser = $agent->browser(Request::header('User-Agent'));
        $version = $agent->version($browser);

        $minAcceptableVersions = [
            'Safari' => '10',
            'IE' => '11',
        ];

        if (in_array($browser, array_keys($minAcceptableVersions)) && (int) $version < $minAcceptableVersions[$browser]) {
            if (Request::route()->getName() !== 'old.browser') {
                return redirect()->route('old.browser');
            }
        } elseif (Request::route()->getName() === 'old.browser') {
            return redirect()->route('clientdash.home');
        }

        return $next($request);
    }
}
```

Controllers

```php
<?php

namespace TIS\Http\Controllers\ClientDashboard;

use ConsoleTVs\Invoices\Classes\Invoice;
use Illuminate\Support\Facades\Auth;
use TIS\Http\Controllers\Controller;
use Facades\TIS\Services\PDFInvoice;
use TIS\Worksheet;

class InvoiceController extends Controller
{
    public function downloadInvoice($id)
    {
        $user = Auth::user();
        $worksheet = $user->isManager() ? Worksheet::findOrFail($id) : $user->worksheets()->findOrFail($id);
        $invoice = PDFInvoice::generateForWorksheet($worksheet);
        abort_unless($invoice instanceof Invoice, 404);
        return $invoice->show();
    }
}
```

Auth & Security

CSRF Protection

```php
<?php

namespace TIS\Http\Middleware;

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

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

```
{{ csrf_field() }}

@javascript('csrfToken', csrf_token())
```

```javascript
import axios from 'axios';

axios.defaults.headers.common = {
    'X-CSRF-TOKEN': window && window.TIS && window.TIS.csrfToken ? window.TIS.csrfToken : null,
    'X-Requested-With': 'XMLHttpRequest',
};

export default axios;
```

```php
<?php

namespace TIS\Http\Middleware;

use Illuminate\Http\Request;
use Fideloper\Proxy\TrustProxies as Middleware;

class TrustProxies extends Middleware
{
    /**
     * The trusted proxies for this application.
     *
     * @var array
     */
    protected $proxies = '**';

    /**
     * The current proxy header mappings.
     *
     * @var array
     */
    protected $headers = [
        Request::HEADER_FORWARDED => 'FORWARDED',
        Request::HEADER_X_FORWARDED_FOR => 'X_FORWARDED_FOR',
        Request::HEADER_X_FORWARDED_HOST => 'X_FORWARDED_HOST',
        Request::HEADER_X_FORWARDED_PORT => 'X_FORWARDED_PORT',
        Request::HEADER_X_FORWARDED_PROTO => 'X_FORWARDED_PROTO',
    ];
}
```

Auth & Guarding

```
php artisan make:auth
```

```php
<?php

Auth::check();
Auth::user();
Auth::id();
Auth::guest();
Auth::login($user);
```

```php
<?php

if (Gate::forUser($user)->allows('update-post', $post)) {
    // The user can update the post...
}
```

Encryption

```php
<?php

use Illuminate\Contracts\Encryption\DecryptException;

// encyrpt
encrypt(request('something-secret'));


// decrypt


try {
    $decrypted = decrypt($encryptedValue);
} catch (DecryptException $e) {
    //
}
```

Hashing

```php
<?php

$request->user()->fill([
    'password' => Hash::make($request->newPassword)
])->save();
```

Caching

```php
<?php

Cache::put('key', 'value', $minutes);
Cache::add('key', 'value', $minutes);
Cache::forever('key', 'value');
Cache::forget('key');
Cache::flush();
$value = Cache::get('key', 'default');
```

Events

```php
<?php

// in your application logic
event(Confoo2017::class);

// in your EventServiceProvider

Confoo2017::class => [
  VancouverAudience::class,
],
```

File Storage

```php
<?php

Storage::put('avatars/1', $fileContents);

$url = Storage::url('file1.jpg');

$url = Storage::temporaryUrl(
    'file1.jpg', Carbon::now()->addMinutes(5)
);
```

Notifications

```
$ php artisan make:notification SayHelloVancouver
```

```php
<?php

class SayHelloVancouver extends Notification
{
    use Queueable;

    /**
     * Get the notification's delivery channels.
     *
     * @param  mixed  $notifiable
     * @return array
     */
    public function via($notifiable)
    {
        return ['mail'];
    }

    /**
     * Get the mail representation of the notification.
     *
     * @param  mixed  $notifiable
     * @return \Illuminate\Notifications\Messages\MailMessage
     */
    public function toMail($notifiable)
    {
        return (new MailMessage)
                    ->line('Hello Vancouver.')
                    ->action('Click here', url('/'))
                    ->line('Thank you for using our application!');
    }
}
```

```php
<?php


// User has Notifiable trait

$user->notify(new SayHelloVancouver());
```

Laravel

Hello!

One of your invoices has been paid!

View Invoice

Thank you for using our application!

Regards,
Laravel

If you're having trouble clicking the "View Invoice" button, copy and paste the URL below into your web browser:

https://example.com/invoice/1

Mail

```
$ php artisan make:mail EmailChanged
```

Jobs

```php
<?php

namespace TIS\Jobs;

use Carbon\Carbon;
use Illuminate\Bus\Queueable;
use Illuminate\Queue\SerializesModels;
use Illuminate\Queue\InteractsWithQueue;
use Illuminate\Contracts\Queue\ShouldQueue;
use Illuminate\Foundation\Bus\Dispatchable;
use Illuminate\Support\Facades\Mail;
use TIS\Employee;
use TIS\Mail\DailyReportEmail;
use Facades\TIS\Services\DailyReportGenerator;

class DailyReport implements ShouldQueue
{
    use Dispatchable, InteractsWithQueue, Queueable, SerializesModels;

    protected $employeesToIncludeInReport = [
        6, // daniel
        82, // joey
    ];

    public function handle()
    {
        $date = Carbon::yesterday();
        $reportData = DailyReportGenerator::data();

        foreach ($this->employeesToIncludeInReport as $employeeId) {
            $employee = Employee::findOrFail($employeeId);
            Mail::to($employee->email)->send(new DailyReportEmail($employee, $date, $reportData));
        }
    }
}
```

```php
<?php

protected function schedule(Schedule $schedule)
{
    $schedule->command('horizon:snapshot')->everyFiveMinutes();
    $schedule->job(DailyReport::class)->dailyAt('00:30')->timezone(config('app.timezone'));
}
```

Queues

```php
<?php

HandleWorksheetFeedback::dispatch($worksheet);
```

```php
<?php

Queue::assertPushed(HandleWorksheetFeedback::class, function ($job) use ($worksheet) {
    return $job->worksheet->id === $worksheet->id;
});
```

Laravel Horizon

- Dashboard
- Monitoring
- Metrics
- Recent Jobs
- Failed

Overview

JOBS PER MINUTE	JOBS PAST HOUR	FAILED JOBS PAST HOUR	STATUS
0	4	0	✓ Active

TOTAL PROCESSES	MAX WAIT TIME	MAX RUNTIME	MAX THROUGHPUT
40	-	client-emails	client-emails

Current Workload

Queue	Processes	Jobs	Wait
client-emails	10	0	A Few Seconds
default	10	0	A Few Seconds
employee-emails	10	0	A Few Seconds
searchable	10	0	A Few Seconds

tis-worker01-OwoD

Supervisor	Processes	Queues	Balancing
supervisor-1	40	default,client-emails,employee-emails,searchable	✓ (Auto)

At the end of the day, it's all just PHP, so you can still deviate when it makes sense to, and create your own architecture

▼ 📁 Services
  ⓒ AccountBalance.php
  ⓒ AuthorizePaypalForClient.php
  ⓒ AuthorizeStripeForClient.php
  ⓒ Cart.php
  ⓒ CreateActivityNote.php
  ⓒ CreateCartItem.php
  ⓒ CreateClient.php
  ⓒ CreateWorksheet.php
  ⓒ CreateWorksheetTally.php
  ⓒ Culler.php
  ⓒ DailyReportGenerator.php
  ⓒ DetermineDiscount.php
  Ⓘ DeterminesTallyAttributesInterface.php
  ⓒ DetermineWorksheetTallies.php
  ⓒ EditorPay.php
  ⓒ PDFInvoice.php
  ⓒ Pricing.php
  ⓒ WorksheetPayment.php

Collections

```php
<?php

$cookie = collect($registrationPage->headers->getCookies())->first(function ($cookie) {
  return $cookie->getName() === 'tis_referral_code';
});
```

https://adamwathan.me/refactoring-to-collections/

String and Array Helpers

```php
<?php


$is_inc = ends_with(data_get($user, 'company', 'No Company Assigned'), ['Inc', 'inc']);
```

Testing

```php
<?php

/** @test */
public function should_format_turnaround()
{
    $this->assertEquals('STANDARD', formatTurnaround('standard'));
    $this->assertEquals('RUSH', formatTurnaround('rush'));
    $this->assertEquals('RUSH+', formatTurnaround('rushplus'));
}
```

```php
<?php

class PDFInvoiceTest extends TestCase
{
    use CreatesWorksheets, RefreshDatabase, MatchesSnapshots, RequestsToClient, AuthenticatesAsClient;

    /** @test */
    public function generates_invoice_with_expected_data()
    {
        $client = factory(Client::class)->create([
            'id' => 501,
            'first_name' => 'Frank',
            'last_name' => 'Jones',
            // ...
        ]);
        $paymentMethod = factory(PaymentMethod::class)->create([
            'client_id' => $client->id,
            'id' => 201,
            'type' => 'visa',
            'last4' => 5401
        ]);
        $worksheet = $this->createEditingWorksheet($client, [
            'payment_method_id' => $paymentMethod->id,
        ]);

        $invoice = PDFInvoice::generateForWorksheet($worksheet);
        $this->assertMatchesSnapshot($invoice->items->toArray());
        $this->assertMatchesSnapshot($invoice->customer_details->toArray());
        $this->assertMatchesSnapshot($invoice->business_details->toArray());
        $this->assertMatchesSnapshot($invoice->transactions->each->forget('date')->toArray());

        $invoice = collect($invoice)->forget('items')->forget('worksheet_number')->forget('name')->forget('number')->forget('customer_details')->forget('business_details')-
>forget('transactions')->forget('date')->toArray();
        $this->assertMatchesSnapshot($invoice);
    }
}
```

```php
<?php

class InternalSPATest extends TestCase
{
    use RefreshDatabase, AuthenticatesAsEmployee, UsesDashboardUrls;

    /** @test */
    function loads_spa_page_with_expected_data()
    {
        $resp = $this->loginAsEmployee()->get($this->internalDashboardUrl());

        $resp
            ->assertStatus(200)
            ->assertViewHas('title', 'TIS Internal Dashboard')
            ->assertViewHas('jwt');
        // ...
    }
```

```php
<?php

public function loginAsEmployee()
{
  $employedEmployee = factory(Employee::class)->states('employed')->create();
  return $this->actingAs($employedEmployee, 'employee');
}
```

```php
<?php

public function rejects_invalid_credentials_but_accepts_valid_ones()
{
    $client = factory(Client::class)->create([
        'email' => 'johndoe@example.net',
        'password' => bcrypt('secret12345'),
    ]);
    $this->browse(function ($browser) use ($client) {
        $browser->visit(new LoginPage)
            // No email or password provided
            ->click('@submit')
            ->assertSee('THE EMAIL FIELD IS REQUIRED.')
            ->assertSee('THE PASSWORD FIELD IS REQUIRED.')
            // Email/pw mismatch due to wrong password
            ->type('@email', 'johndoe@example.net')
            ->type('@password', 'secret1234')
            ->click('@submit')
            ->assertSee('THESE CREDENTIALS DO NOT MATCH OUR RECORDS.')
            // Email/pw mismatch due to wrong email
            ->type('@email', 'johndo@example.net')
            ->type('@password', 'secret12345')
            ->click('@submit')
            ->assertSee('THESE CREDENTIALS DO NOT MATCH OUR RECORDS.')
            // Invalid email format and password too short
            ->type('@email', 'invalidemail')
            ->type('@password', 'short')
            ->click('@submit')
            ->assertSee('THE EMAIL MUST BE A VALID EMAIL ADDRESS.')
            ->assertSee('THE PASSWORD MUST BE AT LEAST 6 CHARACTERS.')
            // Valid email and password combo
            ->type('@email', 'johndoe@example.net')
            ->type('@password', 'secret12345')
            ->click('@submit')
            ->assertPathIs('/')
            ->assertSee(', ' . $client->first_name . '!');
    });
}
```

Front-end

Blade (templating)

Laravel Mix (webpack wrapper)

Presets (for js frameworks)

Ecosystem

Packages

Laravel Valet

Laravel Homestead

Laravel Horizon

Laravel Cashier

Laravel Socialite

Lumen

Spark

Laravel Scout

Laravel Passport

Laravel Mix

https://github.com/topics/laravel and
https://github.com/topics/laravel-package

https://spatie.be/en/opensource/laravel

Third Party Services

Bug Tracking and Monitoring

Community

CHICAGO, IL

LARACON

July 25 - 26 2018

LARACON
ONLINE

Experience Laracon
All Around the World

Join 6,000+ Laravel Developers on February 7, 2018

Early Bird Registration
Just $12

Tickets go to $25 on January 3rd, 2018

https://larachat.co/
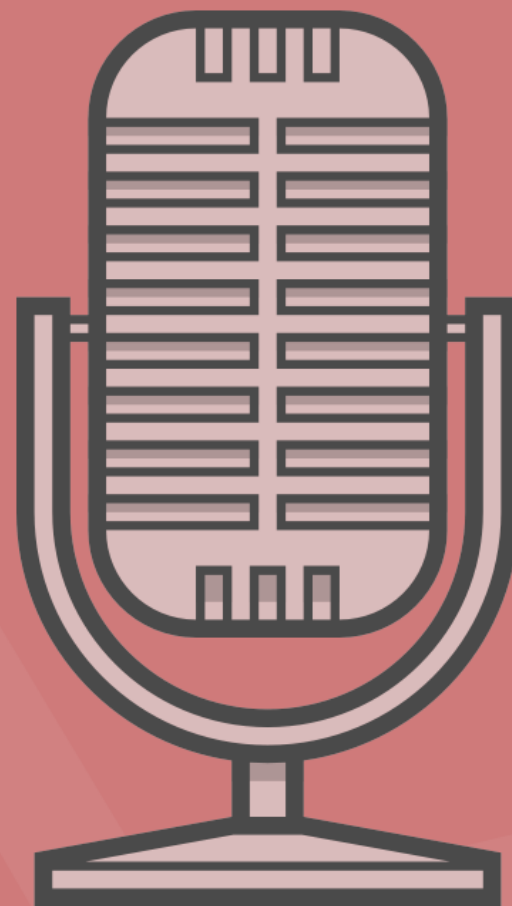
Laravel Podcast
Season Three
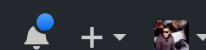
The Laravel PHP Framework

http://laravel.com

📖 **Repositories** 32   👥 People 11

Pinned repositories

laravel
A PHP Framework For Web Artisans

🟣 PHP   ⭐ 36.6k   ⑂ 12.1k

framework

🟣 PHP   ⭐ 8.7k   ⑂ 4.9k

docs

⭐ 1.1k   ⑂ 2.4k

Search repositories...   Type: All ▾   Language: All ▾

framework

php   laravel   framework

🟣 PHP   ⭐ 8,673   ⑂ 4,932   ⚖ MIT   Updated 4 hours ago

docs

⭐ 1,117   ⑂ 2,425   ⚖ MIT   Updated 4 hours ago

laravel

A PHP Framework For Web Artisans

php   laravel   framework

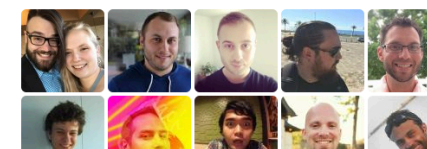🟣 PHP   ⭐ 36,591   ⑂ 12,066   Updated 10 hours ago

Top languages

🟣 PHP   🟢 Shell   🟡 JavaScript
🔵 TypeScript

Most used topics

framework   laravel   php

People   11 ›

Thank You & Q&A

https://slides.jkudish.com