Building "complex" applications with **Laravel**®

About me

- 10+ years in Gaumina

- 3 years in Sheology

- Developer, IT consultant, analyst, manager, business development director, project manager, sales manager, programming director, Darth Vader, The White Wolf,

  … etc.

define complex application

- CMS for CMS-es
- Decoupled (componential)
- Sophisticated frontend
- Instant / async execution
- Extendable / updateable
- Package-based
- Integrate-able with external APIs
- … using PHP and JavaScript

SES (Sheology Enterprise System)

- Laravel 4.2 base, Angular 1.x for front-end interactions
- 15 separate components
- CMS integration (WordPress, BB4W)
- CommandBus + Queues + Beanstalkd + Supervisor
- git version control
- 30+ composer, 25+ bower external packages
- Facebook Graph API, Facebook Marketing API, Twitter API integration
- gulp tasks for production deployment
- … PHP and JavaScript based

SES (Sheology Enterprise System)

- 50 system users, 10 requests per minute

- 4,000,000 active users per month through integrations

- 1,642 commits

- 47.8% JavaScript*, 44.2% PHP, 7.6% CSS, 0.1% HTML

* including minified versions

Patterns / methods used

- Service – Repository pattern
  http://heera.it/laravel-repository-pattern

- CommandBus pattern (adopted)
  https://github.com/laracasts/Commander

- Queues + Beanstalkd + Supervisor
  https://laravel.com/docs/5.1/queues

- External API SDKs
  https://github.com/facebook/facebook-php-sdk-v4
  https://github.com/facebook/facebook-php-ads-sdk
  https://github.com/googleads/googleads-php-lib

General guidelines

to make your ~~coding~~ life easier
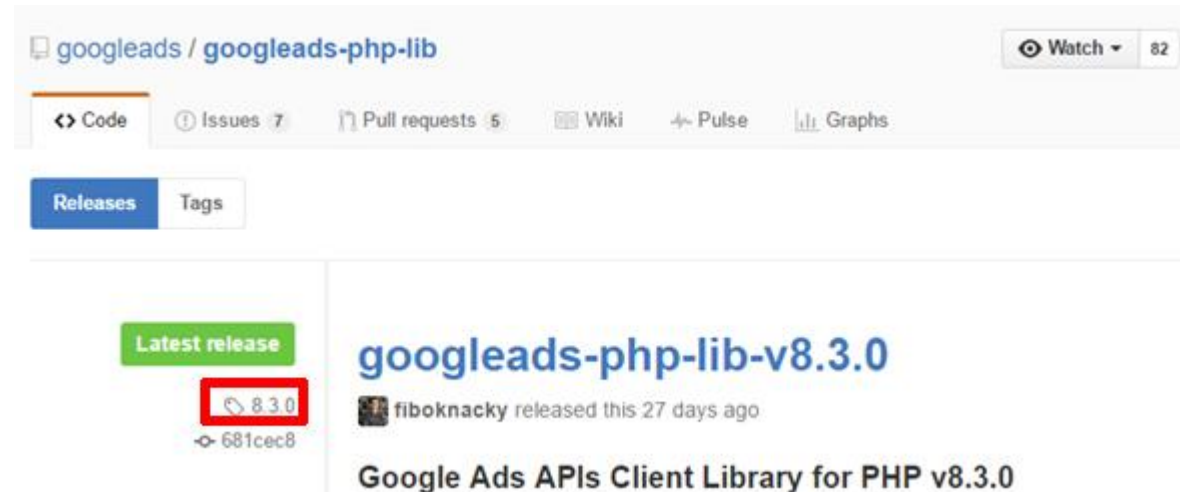
General guidelines

- Don't use 3rd party wrappers for official SDKs

General guidelines

- Look for <u>really</u> suitable pattern usages

"ORM Repository is cool, because I may want to replace Eloquent with MongoDB some day"

A "Repository" is a class, which allows grouping commonly used queries into methods.

General guidelines

- Re-use patterns out of their declared scope, where suitable

```php
1   <?php namespace MyApp\Utils;
2
3   use Facebook\Facebook;
4
5   /**
6   * @class FacebookRepository Our repository, containing commonly used queries
7   */
8   class FacebookRepository
9   {
10          /** @var Facebook */
11          protected $fb;
12
13          public function __construct()
14          {
15                  $this->fb = new Facebook([
16                          'app_id'             => $_ENV['facebook.app_id'],
17                          'app_secret'         => $_ENV['facebook.app_secret'],
18                          'default_graph_version' => $_ENV['facebook.version'],
19                  ]);
20          }
21  }
```

FacebookRepository.php hosted with ❤ by GitHub   view raw

Facebook SDK Repository Class

https://medium.com/@YOzaz/facebook-php-sdk-repository-bfa374fe58d4

General guidelines

- Automate everything!

```
1   var gulp = require('gulp');
2   var concat = require('gulp-concat');
3   var ngAnnotate = require('gulp-ng-annotate');
4   var plumber = require('gulp-plumber');
5
6   gulp.task('app', function() {
7       return gulp.src(['public/app/**/app.js', 'public/app/**/*.module.js', 'public/app/**/*.js'])
8               .pipe(plumber())
9                       .pipe(concat('app.js', {newLine: ';'}))
10                      .pipe(ngAnnotate({add: true}))
11              .pipe(plumber.stop())
12          .pipe(gulp.dest('public/src/'));
13  });
```

gulpfile.js hosted with ♥ by GitHub   view raw

General guidelines

- Combine packages

```php
/**
 * Syncs subscription process
 *
 * @class MailChimp
 *
 * @param string $list_id
 * @param string $email
 */
public function syncSubscribe( $list_id, $email )
{
    $audience = $this->audience->getBySource( $list_id, self::AUDIENCE_TYPE_MAILCHIMP );

    if ( !$audience )
    {
        return;
    }

    $this->facebook->addEmailToAudience( $email, $audience );
}
```

General guidelines

- Decouple component execution

name	current-jobs-urgent	current-jobs-ready	current-jobs-reserved	current-jobs-delayed	current-jobs-buried	total-jobs
default	0	0	0	0	0	0
dfp	0	0	0	0	0	700
facebookads	0	0	0	0	0	93
mailer	0	0	0	0	0	0
newsletter	0	0	0	5	0	383
publisher	0	0	0	87	0	21337

Beanstalk console   localhost:11300 ▾   All tubes ▾   Toolbox ▾   Links ▾

General guidelines

- Make your code readable (for yourself)

```php
/**
 * Show the form for creating a new article
 *
 * @return \Illuminate\Http\Response
 */
public function create()
{
    $entity = Entity::getByTypeForEdit('article');
    $meta_entity = Entity::getByTypeForEdit('meta');
    $publishers = Publisher::getPublishersForEdit( $entity );
    $publishers_enabled = $this->getEnabledTemplates( 'article' );

    $this->displayView('articles.create', compact('publishers','entity','meta_entity','publishers_enabled'));
}
```

General guidelines

- Skinny controllers & models, fat Services > Commands > Repositories

```php
/**
 * Updates SES campaign group
 * Automatically resolves type, and updates binding group as well
 *
 * @param int $id
 * @param array $data
 *
 * @return \SES\Entities\Commercial\Ses\Group\SesCampaignGroup
 */
public function update( $id, $data )
{

    $command = new UpdateSesCampaignGroupCommand( $data, $id );
    $ses_campaign_group = $this->commandBus->execute( $command );

    return $ses_campaign_group;

}
```

General guidelines

- Create "base" classes for future extensions


```php
abstract class Content extends Eloquent
{
    /**
     * Access entity from model
     */
    use EntityAccess;

    /**
     * Access repository and service from model
     */
    use RepositoryServiceAccess;

    /**
     * Soft deletes
     */
    use SoftDeletingTrait;

    // Validation rules here
    public $rules = [
        'title'     => 'required',
    ];
```

```php
class BaseController extends Controller {

    /**
     * Setup the layout used by the controller.
     *
     * @return void
     */
    protected function setupLayout()
    {

        if ( !is_null( $this->layout ))
        {
            $this->layout = View::make( $this->layout );
        }
    }

    /**
     * Display view
     *
     * @param $view
     * @param array $data
     */
    protected function displayView( $view, array $data = [] )
    {
        $this->layout->content = View::make( $view, $data );
    }
}
```

General guidelines

- Sort by purpose, not by type
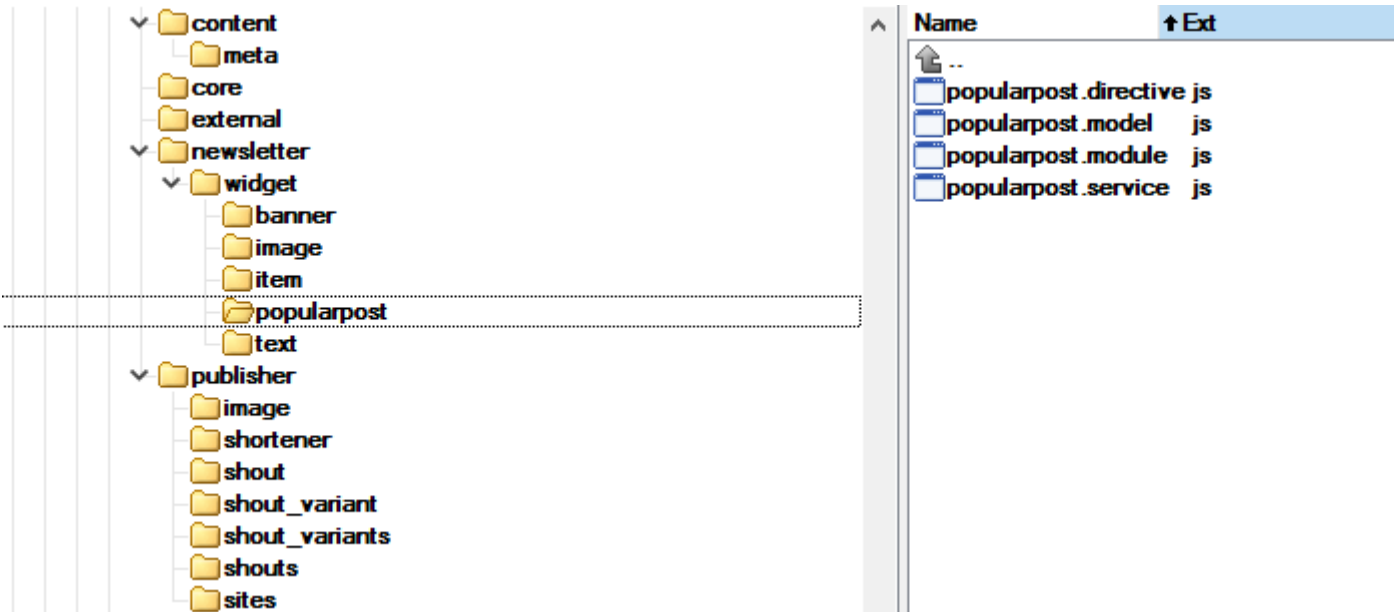
General guidelines

- Create as many methods as you need*

```php
/**
 * Gets image object by url
 *
 * @param string $url image url
 * @return Image
 **/
public function getByUrl( $url )
{
    return $this->model->ofUrl($url)->first();
}
```

```php
/**
 * Gets image by Url and version
 *
 * @param string $url image url
 * @param int $version_id
 * @return Image
 **/
public function getByUrlAndVersion( $url, $version_id = null )
{
    if ( !isset($version_id) )
    {
        return $this->getByUrl( $url );
    }
    else
    {
        return $this->model->ofUrl($url)->ofVersion($version_id)->first();
    }
}
```

* and name them by purpose

General guidelines

- Be production-safe, but development-informative

```php
public function thisIsAVeryCoolMethod( $data = [] )
{
    try
    {
        /** some sophisticated code */
    }
    catch ( Exception $e )
    {
        if ( \App::environment('local') )
        {
            throw $e;
        }

        \Log::error( $e->getMessage() );
        return false;
    }

    return true;
}
```

Tips & Tricks

'cause we're all hackers

Tips & Tricks

- Use attribute getters for constant model values

```php
/**
 * Is targeting set to mobile attribute
 */
public function getMobileAttribute()
{
    $placement = $this->placement;

    if ( !isset($placement) )
    {
        return null;
    }

    return $placement->mobile;
}
```

```php
/**
 * Is targeting set to mobile attribute
 */
public function getMobilePlacementAttribute()
{
    $targeting = $this->targeting;

    return isset($targeting) ? $targeting->mobile : false;
}
```

Tips & Tricks

- Little helpers that makes the difference...

```php
/**
 * Lazy loads relationship if required
 *
 * @param \Eloquent $model
 * @param string $relationship
 *
 * @return mixed
 */
function load( &$model, $relationship )
{
    if ( !isset($model) )
    {
        return null;
    }

    if ( !isset( $model->{$relationship} ) )
    {
        $model->load( $relationship );
    }

    return $model->{$relationship};
}
```

```php
// Angular Wrap
function aw( $statement )
{
    return '{{' . $statement . '}}';
}
```

```php
/**
 * Escapes text for angular argument
 *
 * @param string $string
 *
 * @return string
 */
function escape_for_angular( $string )
{
    return str_replace( "'", "\\'", htmlentities( $string ) );
}
```

Tips & Tricks

- Use illegal tricks to speed up your debugging

```php
class UserRepository extends AbstractBaseRepository
{
    /** ... */

    /**
     * @return mixed
     **/
    public function __call( $method, $args )
    {
        return call_user_func_array( [$this->model, $method], $args );
    }

    public function customRepositoryMethod()
    {
    }
}

(new UserRepository)->customRepositoryMethod(); //works
(new UserRepository)->find(1); // works
```

Tips & Tricks

- Don't rely on cache

```
for ( $i = 0; $i < 10000000; $i++ )
{
    $user = User::getWithLotsOfRelationshipsAndUnindexedSQLsCached( Input::get('id') );
}
```

… or server administrator may kill you.

Tips & Tricks

- Log model author (Laravel 4.x)... with additional fields if required

```php
static::creating( function($model)
{
    $user = User::current();

    if ( isset($user) )
    {
        $model->created_by = $user->id;
        $model->updated_by = $user->id;
    }
});
```

```php
static::updating( function($model)
{
    $user = User::current();

    if ( isset($user) )
    {
        $model->updated_by = $user->id;

        $dirty = $model->getDirty();

        if ( isset($dirty['published_at']) )
        {
            $model->published_by = $user->id;
        }
    }
});
```

Tips & Tricks

- Ensure argument is an array*

```
/**
 * Scope for searching entities for specified model
 *
 * @param \Illuminate\Database\Query\Builder $query
 * @param string|array $type
 *
 * @return \Illuminate\Database\Query\Builder
 */
public function scopeOfType($query, $type)
{
    $type = Collection::make( $type )->all();

    return $query->whereIn( 'type', $type );
}
```

* so it accepts an array, non-array, or \Illuminate\Support\Collection

Tips & Tricks

- Ultra-powered CSRF checker*

```php
public function filter()
{
    // Wrap all requests in the CSRF filter.  This
    // makes sure all non-GET requests have a valid CSRF.
    $method = Request::getMethod();
    $token = Session::token();
    if ( $method !== 'GET'
        && $method !== 'HEAD'
        && Input::get('_token') !== $token // Checks form submit
        && Request::header('x-csrf') !== $token // Checks AJAX
    ) {
        // Show error screen
        throw new TokenMismatchException;
    }
}
```

* because built-in s*cks.

Tips & Tricks

- Database reconnection for queues*

```
try
{
    DB::connection()->getDatabaseName();
}
catch(\Exception $e)
{
    DB::reconnect();
}
```

* so an admin doesn't kill you (again).

Interesting packages

we're not lazy – we are opensource

Interesting packages

Laravel Verify Bundle

A simple role/permission authentication bundle for Laravel

- Secure password storage with salt
- Role/permission based authentication
- Exceptions for intelligent handling of errors
- Configurable/extendable

https://github.com/Toddish/Verify

Cool, because:

- Extends Eloquent
- Laravel-style
- Maintained
- Really simple to understand

Interesting packages

Laravel Breadcrumbs 3

A simple Laravel-style way to create breadcrumbs in Laravel 5.

```
Breadcrumbs::register('home', function($breadcrumbs) {
    $breadcrumbs->push('Home', route('home'));
});
```

Cool, because:

- Flexible
- Laravel-style
- Maintained
- Simple to understand

https://github.com/davejamesmiller/laravel-breadcrumbs

Interesting packages

Transform PHP Vars to JavaScript

`build` `passing`

Often, you'll find yourself in situations, where you want to pass some server-side string/array/collection/whatever to your JavaScript. Traditionally, this can be a bit of a pain - especially as your app grows.

This package simplifies the process drastically.

https://github.com/laracasts/PHP-Vars-To-Js-Transformer

Interesting packages

Laravel - Assets Versioning

`build passing` `stable 0.3.3` `downloads 1.71 k`

Have you ever had a problem with cache in your assets? This package may help you.

```html
<link rel="stylesheet" href="{{ Asset::v('assets/stylesheets/css/main.css') }}" />
<script src="{{ Asset::v('assets/javascripts/js/main.js') }}"></script>
```

In your **local** environment, nothing changes. But in **production**, you just need to run the following command every time you need to update your assets' version:

```
$ php artisan asset:dist
```

And your layout will be rendered as this:

```html
<link rel="stylesheet" href="/assets/stylesheets/dist/1392745827/main.css" />
<script src="/assets/javascripts/dist/1392745827/main.js"></script>
```

https://github.com/EscapeWork/laravel-asset-versioning

Let's share

for free!

Let's share

```php
/**
 * Replaces remote URL to Local Path
 *
 * @param string $url remote image or file (with http and www stuff)
 *
 * @return string
 */
public function convertToLocalPath( $url )
{
        $app_url = rtrim( Config::get('app.url'), '/' );
        $public_path = rtrim( public_path(), '/' );
        $result = $url;

        $result = str_replace( $app_url, $public_path, $result );

        $app_url = str_replace( 'https://', 'http://', $app_url );
        $result = str_replace( $app_url, $public_path, $result );

        return $result;
}
```

https://gist.github.com/YOzaz/c11140c877d
04732b6147d0277a2595d

Let's share

Laravel and Swift Mailer integrator

Package makes error-safe calls to SwiftMailer transport reset functions before email is being sent. In case reset fails - SMTP connection is restarted, thus maintaining it active for whole application living cycle.

Important for long-living applications, when emails are sent through Beanstalkd + Supervisor + Laravel Queue Daemon Worker architecture.

https://github.com/YOzaz/Laravel-SwiftMailer

Thank you.