




10 less-known Laravel packages

May 2016

Povilas Korop | Laravel Vilnius Meetup
2016-05-11



1. Laravel Generator

<https://github.com/InfyOmLabs/laravel-generator>

InfyOm Laravel Generator - API, Scaffold, CRUD Laravel Generator

CORE GENERATOR

Generates Model, Repository, Requests and Migrations

API GENERATOR

Generates APIs with validations and adjusts routes file

SCAFFOLD GENERATOR

Generates CRUD with validations and adjusts routes file

```
namespace App\Models;

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

class Post extends Model
{
    use SoftDeletes;

    public $table = "posts";

    protected $dates = ['deleted_at'];

    public $fillable = [
        "title",
        "body",
        "author"
    ];

    /** The attributes that should be casted to native types. ... */
    protected $casts = [
        "title" => "string",
        "body" => "string",
        "author" => "string"
    ];

    /** Validation rules ... */
    public static $rules = [
        "title" => "required",
        "author" => "required"
    ];
}
```

2. Laravel Localize Middleware

<https://github.com/BenConstable/laravel-localize-middleware>

Configurable
localization
middleware for your
Laravel application

Determining the locale from a request parameter

Driver name: `parameter`

The default determiner sets the application locale from a request parameter called `locale`. You can change this using the `parameter` configuration option.

This allows you to, for example, set up your application routes like:

```
// app/Http/routes.php

Route::group(['prefix' => '{locale}'], function () {
    // Routes here...
});
```

Determining the locale from a request header

Driver name: `header`

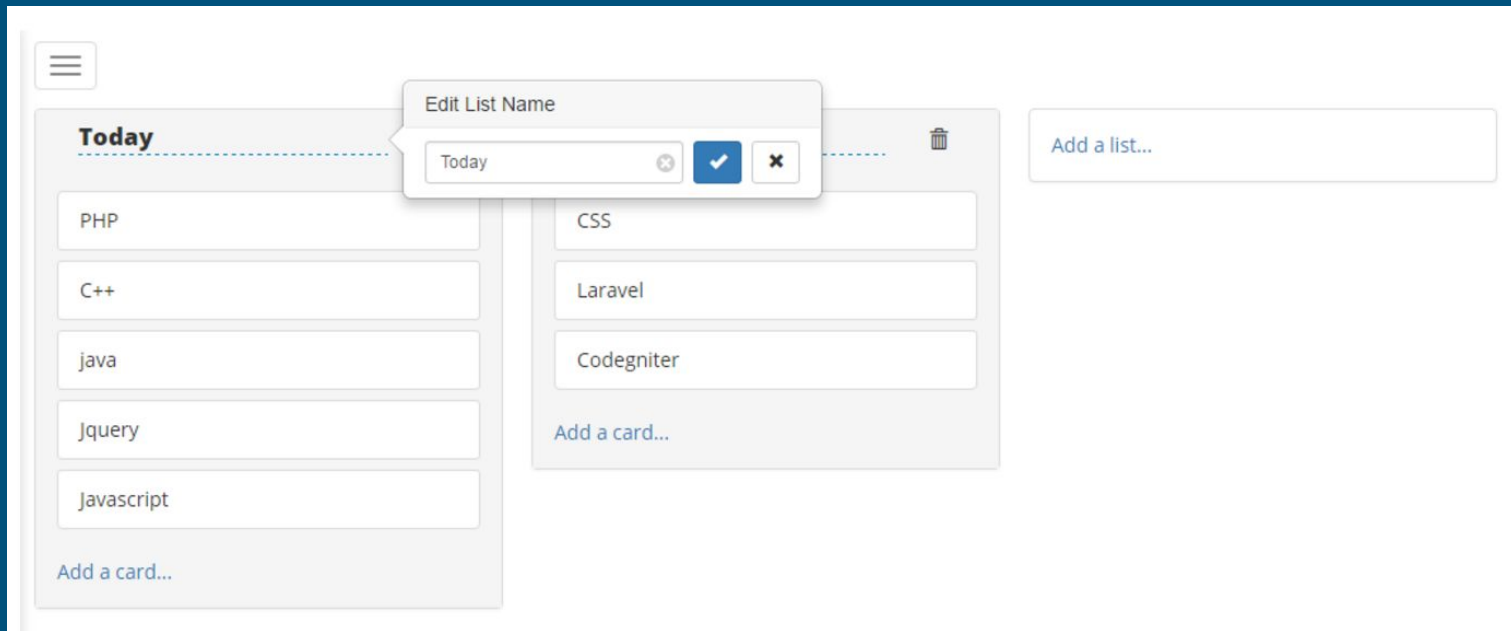
This determiner sets the application locale from a request header, which defaults to `Accept-Language`. You can change this using the `header` configuration option.

Aside: For information on using `Accept-Language` to determine the locale, see [this info from the W3C](#).

3. Dingo project management

<https://github.com/zeeshanu/dingo-project-managment>

Trello-like clone written in Laravel!



4. Multi-tenant

<https://github.com/hyn/multi-tenant>

Run multiple websites using the same laravel installation while keeping tenant specific data separated.

Multi tenancy how?

In its most abstract sense you can use this package to manage multiple websites with only one application installation.

- Multiple websites running on one code base.
- Multiple hostnames configured per website.

Each website has its own folder on disk, allowing:

- separation of routes, templates, translations etc
- custom files (media, themes and packages)

Also each website has its own database, this ensures that in no way one website can access data from another website. The distinction also gives proper division of responsibilities to the system (global) and tenant (local) databases.

5. Teamwork

<https://github.com/mpociot/teamwork>

User to Team associations with invitation system for the Laravel 5 Framework.

```
$team = new Team();
$team->owner_id = User::where('username', '=', 'sebastian')->first()->getKey();
$team->name = 'My awesome team';
$team->save();

$myOtherCompany = new Team();
$myOtherCompany->owner_id = User::where('username', '=', 'marcel')->first()->getKey();
$myOtherCompany->name = 'My other awesome team';
$myOtherCompany->save();
```

Now thanks to the `UserHasTeams` trait, assigning the Teams to the user is uber easy:

```
$user = User::where('username', '=', 'sebastian')->first();

// team attach alias
$user->attachTeam($team); // parameter can be a Team object, array, or id

// or eloquent's original technique
$user->teams()->attach($team->id); // id only
```

6. Laravel Cascade Soft-deletes

<https://github.com/michaeldyrynda/laravel-cascade-soft-deletes>

Cascading deletes for Eloquent models that implement soft deletes

```
use App\Comment;
use Iatstuti\Database\Support\CascadeSoftDeletes;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\SoftDeletes;

class Post extends Model
{
    use SoftDeletes, CascadeSoftDeletes;

    protected $cascadeDeletes = ['comments'];

    protected $dates = ['deleted_at'];

    public function comments()
    {
        return $this->hasMany(Comment::class);
    }
}
```

Now you can delete an `App\Post` record, and any associated `App\Comment` records will be deleted as well. Since the `Post` record implements the `CascadeSoftDeletes` trait as well, it's children will also be deleted and soft deleted.

```
$post = App\Post::find($postId)
$post->delete(); // Soft delete the post, which will also trigger the delete()
```

7. Command Validator

<https://github.com/cerbero90/command-validator>

Validate Laravel console commands input.

```
use Illuminate\Console\Command;
use Cerbero\CommandValidator\ValidatesInput;

class Example extends Command
{
    use ValidatesInput;

    public function rules()
    {
        return [
            'year' => 'digits:4|min:2000'
        ];
    }
}
```

```
public function messages()
{
    return [
        'min' => 'The minimum allowed :attribute is :min'
    ];
}

public function attributes()
{
    return [
        'year' => 'year of birth'
    ];
}
```


8. Laravel Server Monitor

<https://github.com/ericmakesstuff/laravel-server-monitor>

Server Monitoring Command for Laravel Applications

```
php artisan monitor:run
```

You can run only certain monitors at a time:

```
php artisan monitor:run HttpPing  
php artisan monitor:run SSLCertificate,DiskUsage
```

Alert Configuration

Alerts can be logged to the default log handler, or sent via email, Pushover, or Slack. Allowed values are `log`, `mail`, `pushover`, and `slack`.

The default alert configurations are:

```
'events' => [  
    'whenDiskUsageHealthy'    => ['log'],  
    'whenDiskUsageAlarm'     => ['log', 'mail'],  
    'whenHttpPingUp'         => ['log'],  
    'whenHttpPingDown'       => ['log', 'mail'],  
    'whenSSLCertificateValid' => ['log'],  
    'whenSSLCertificateInvalid' => ['log', 'mail'],  
    'whenSSLCertificateExpiring' => ['log', 'mail'],  
],
```

9. Laravel Langman

<https://github.com/themsaid/laravel-langman>

Language files manager in your artisan console.

```
done!

~/sites/laravel-sample on master
$ php artisan langman:show users
+-----+-----+-----+
| key   | ar   | en   |
+-----+-----+-----+
| country | Dawla | Country |
| name.first | esm awal | first name |
| age     | 3omr  | age     |
| type    | noo3  | type    |
+-----+-----+-----+

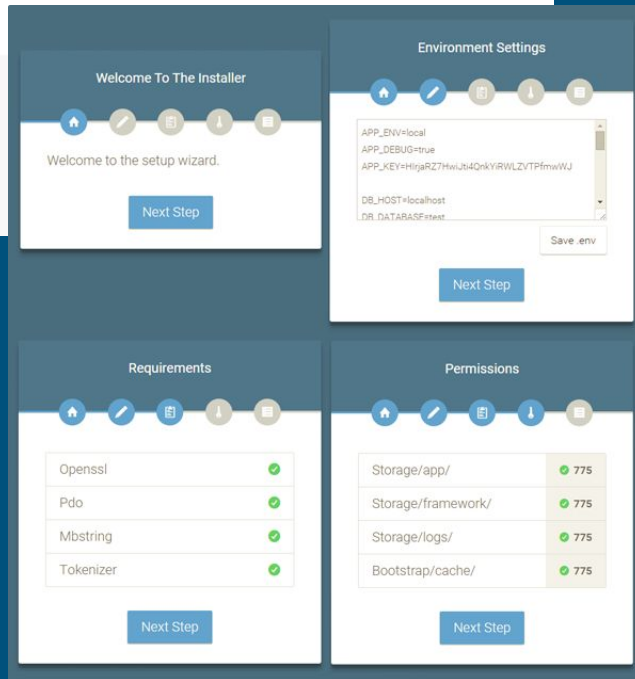
~/sites/laravel-sample on master
$
```

10. Laravel Web Installer

<https://github.com/RachidLaasri/LaravelInstaller>

Do you want your clients to be able to install a Laravel project just like they do with WordPress or any other CMS? This Laravel package allows users who don't use Composer, SSH etc to install your application just by following the setup wizard. The current features are :

- Check For Server Requirements.
- Check For Folders Permissions.
- Ability to set database information.
- Migrate The Database.
- Seed The Tables.



Thank you.