
Laraplans Documentation

Release 2.1.0

Gerardo J. Báez

May 15, 2019

Contents

1	Installation	3
1.1	Composer	3
1.2	Service Provider	3
1.3	Config File and Migrations	3
1.4	Traits and Contracts	4
2	Usage	5
2.1	Create a Plan	5
2.2	Accessing Plan Features	5
3	Events	7

This library intends to provide a quick starting point to any app that requires SaaS style plans and features.

CHAPTER 1

Installation

1.1 Composer

```
$ composer require rayafort/plans
```

1.2 Service Provider

Add `RayaFort\Plans\PlansServiceProvider::class` to your application service providers file: `config/app.php`.

```
'providers' => [  
    /**  
     * Third Party Service Providers...  
     */  
    RayaFort\Plans\PlansServiceProvider::class,  
]
```

1.3 Config File and Migrations

Publish package config file and migrations with the following command:

```
php artisan vendor:publish --provider="RayaFort\Plans\PlansServiceProvider"
```

Then run migrations:

```
php artisan migrate
```

1.4 Traits and Contracts

Add `RayaFort\Plans\Traits\PlanSubscriber` trait and `RayaFort\Plans\Contracts\PlanSubscriberInterface` contract to your `User` model.

See the following example:

```
namespace App\Models;

use Illuminate\Foundation\Auth\User as Authenticatable;
use RayaFort\Plans\Contracts\PlanSubscriberInterface;
use RayaFort\Plans\Traits\PlanSubscriber;

class User extends Authenticatable implements PlanSubscriberInterface
{
    use PlanSubscriber;
```


2.1 Create a Plan

```
use RayaFort\Plans\Models\Plan;
use RayaFort\Plans\Models\PlanFeature;

$plan = Plan::create([
    'name' => 'Pro',
    'description' => 'Pro plan',
    'price' => 9.99,
    'interval' => 'month',
    'interval_count' => 1,
    'trial_period_days' => 15,
    'sort_order' => 1,
]);

$plan->features()->saveMany([
    new PlanFeature(['code' => 'listings', 'value' => 50, 'sort_order' => 1]),
    new PlanFeature(['code' => 'pictures_per_listing', 'value' => 10, 'sort_order' => 5]),
    new PlanFeature(['code' => 'listing_duration_days', 'value' => 30, 'sort_order' => 10]),
    new PlanFeature(['code' => 'listing_title_bold', 'value' => 'Y', 'sort_order' => 15])
]);
```

2.2 Accessing Plan Features

In some cases you need to access a particular feature in a particular plan, you can accomplish this by using the `getFeatureByCode` method available in the `Plan` model.

Example:

```
$feature = $plan->getFeatureByCode('pictures_per_listing');  
$feature->value // Get the feature's value
```

The following are the events fired by the package:

- `RayaFort\Plans\Events\SubscriptionCreated`: Fired when a subscription is created.
- `RayaFort\Plans\Events\SubscriptionRenewed`: Fired when a subscription is renewed using the `renew()` method.
- `RayaFort\Plans\Events\SubscriptionCanceled`: Fired when a subscription is canceled using the `cancel()` method.
- `RayaFort\Plans\Events\SubscriptionPlanChanged`: Fired when a subscription's plan is changed; it will be fired once the `PlanSubscription` model is saved. Plan change is determined by comparing the original and current value of `plan_id`.