

Prologue

Release Notes

• Upgrade Guide

Contribution Guide

API Documentation

Getting Started

Architecture Concepts

The Basics

Frontend

Security

Digging Deeper

Database

Eloquent ORM

Testing

Official Packages



Deploy Laravel and PHP applications on DigitalOcean, Linode, & AWS.

ADS VIA CARBON

Upgrade Guide

Upgrading To 6.0 From 5.8

High Impact Changes

- [Authorized Resources & viewAny](#)
- [String & Array Helpers](#)

Medium Impact Changes

- [Carbon 1.x No Longer Supported](#)
- [Redis Default Client](#)
- [Database Capsule::table Method](#)
- [Eloquent Arrayable & toArray](#)
- [Eloquent BelongsTo::update Method](#)
- [Eloquent Primary Key Types](#)
- [Localization Lang::trans and Lang::transChoice Methods](#)
- [Localization Lang::getFromJson Method](#)
- [Queue Retry Limit](#)
- [Resend Email Verification Route](#)
- [The Input Facade](#)

Upgrading To 6.0 From 5.8

Estimated Upgrade Time: One Hour



We attempt to document every possible breaking change.

Since some of these breaking changes are in obscure parts of the framework only a portion of these changes may actually affect your application.

PHP 7.2 Required

Likelihood Of Impact: Medium

PHP 7.1 will no longer be actively maintained as of December 2019. Therefore, Laravel 6.0 requires PHP 7.2 or greater.

Updating Dependencies

Update your `laravel/framework` dependency to `^6.0` in your `composer.json` file.

Next, examine any 3rd party packages consumed by your application and verify you are using the proper version for Laravel 6 support.

Authorization

Authorized Resources & viewAny

Likelihood Of Impact: High

Authorization policies attached to controllers using the `authorizeResource` method should now define a `viewAny` method, which will be called when a user accesses the controller's `index` method. Otherwise, calls to the `index` method of the controller will be rejected as unauthorized.

Authorization Responses

Likelihood Of Impact: Low

The constructor signature of the `Illuminate\Auth\Access\Response` class has changed. You should update your code accordingly. If you are not constructing authorization responses manually and are only using the `allow` and `deny` instance methods within your policies, no change is required:

```
/**
 * Create a new response.
 *
 * @param bool $allowed
 * @param string $message
 * @param mixed $code
 * @return void
 */
public function __construct($allowed, $message = '', $code = null)
```

Returning "Deny" Responses

Likelihood Of Impact: Low

In previous releases of Laravel, you did not need to return the value of the `deny` method from your policy methods since an exception was thrown immediately. However, in accordance with the Laravel documentation, you must now return the value of the `deny` method from your policies:

```
public function update(User $user, Post $post)
{
    if (! $user->role->isEditor()) {
        return $this->deny("You must be an editor to edit this post.");
    }

    return $user->id === $post->user_id;
}
```

The `Illuminate\Contracts\Auth\Access\Gate` Contract

Likelihood Of Impact: Low

The `Illuminate\Contracts\Auth\Access\Gate` contract has received a new `inspect` method. If you are implementing this interface manually, you should add this method to your implementation.

Carbon

Carbon 1.x No Longer Supported

Likelihood Of Impact: Medium

Carbon 1.x [is no longer supported](#) since it is nearing its maintenance end of life. Please upgrade your application to Carbon 2.0.

Configuration

The `AWS_REGION` Environment Variable

Likelihood Of Impact: Optional

If you plan to utilize [Laravel Vapor](#), you should update all occurrences of `AWS_REGION` within your `config` directory to `AWS_DEFAULT_REGION`. In addition, you should update this environment variable's name in your `.env` file.

Redis Default Client

Likelihood Of Impact: Medium

The default Redis client has changed from `redis` to `phpredis`. In order to keep using `redis`, ensure the `redis.client` configuration option is set to `redis` in your `config/database.php` configuration file.

Database

The Capsule `table` Method

Likelihood Of Impact: Medium



This change only applies to non-Laravel applications that are using `illuminate/database` as a dependency.

The signature of the `Illuminate\Database\Capsule\Manager` class' `table` method has updated to accept a table alias as its second argument. If you are using `illuminate/database` outside of a Laravel application, you should update any calls to this method accordingly:

```
/**
 * Get a fluent query builder instance.
```

```

*
* @param \Closure|\Illuminate\Database\Query\Builder|string $table
* @param string|null $as
* @param string|null $connection
* @return \Illuminate\Database\Query\Builder
*/
public static function table($table, $as = null, $connection = null)

```

The `cursor` Method

Likelihood Of Impact: Low

The `cursor` method now returns an instance of `Illuminate\Support\LazyCollection` instead of a `Generator`. The `LazyCollection` may be iterated just like a generator:

```

$users = App\User::cursor();

foreach ($users as $user) {
    //
}

```

Eloquent

The `BelongsTo::update` Method

Likelihood Of Impact: Medium

For consistency, the `update` method of the `BelongsTo` relationship now functions as an ad-hoc update query, meaning it does not provide mass assignment protection or fire Eloquent events. This makes the relationship consistent with the `update` methods on all other types of relationships.

If you would like to update a model attached via a `BelongsTo` relationship and receive mass assignment update protection and events, you should call the `update` method on the model itself:

```

// Ad-hoc query... no mass assignment protection or events...
$post->user()->update(['foo' => 'bar']);

// Model update... provides mass assignment protection and events...
$post->user->update(['foo' => 'bar']);

```

Arrayable & `toArray`

Likelihood Of Impact: Medium

The Eloquent model's `toArray` method will now cast any attributes that implement `Illuminate\Contracts\Support\Arrayable` to an array.

Declaration Of Primary Key Type

Likelihood Of Impact: Medium

Laravel 6.0 has received [performance optimizations](#) for integer key types. If you are using a string as your model's primary key, you should declare the key type using the `$keyType` property on your model:

```

/**
 * The "type" of the primary key ID.
 *
 * @var string
 */
protected $keyType = 'string';

```

Email Verification

Resend Verification Route HTTP Method

Likelihood Of Impact: Medium

To prevent possible CSRF attacks, the `email/resend` route registered by the router when using Laravel's built-in email verification has been updated from a `GET` route to a `POST` route. Therefore, you will need to update your frontend to send the proper request type to this route. For example, if you are using the built-in email verification template scaffolding:

```

{{ __('Before proceeding, please check your email for a verification link.') }}
{{ __('If you did not receive the email') }},

<form class="d-inline" method="POST" action="{{ route('verification.resend') }}">

```

```
@csrf

<button type="submit" class="btn btn-link p-0 m-0 align-baseline">
    {{ __('Click here to request another') }}
</button>

</form>
```

The `MustVerifyEmail` Contract

Likelihood Of Impact: Low

A new `getEmailForVerification` method has been added to the `Illuminate\Contracts\Auth\MustVerifyEmail` contract. If you are manually implementing this contract, you should implement this method. This method should return the object's associated email address. If your `App\User` model is using the `Illuminate\Auth\MustVerifyEmail` trait, no changes are required, as this trait implements this method for you.

Helpers

String & Array Helpers Package

Likelihood Of Impact: High

All `str_` and `array_` helpers have been moved to the new `laravel/helpers` Composer package and removed from the framework. If desired, you may update all calls to these helpers to use the `Illuminate\Support\Str` and `Illuminate\Support\Arr` classes. Alternatively, you can add the new `laravel/helpers` package to your application to continue using these helpers:

```
composer require laravel/helpers
```

Localization

The `Lang::trans` & `Lang::transChoice` Methods

Likelihood Of Impact: Medium

The `Lang::trans` and `Lang::transChoice` methods of the translator have been renamed to `Lang::get` and `Lang::choice`.

In addition, if you are manually implementing the `Illuminate\Contracts\Translation\Translator` contract, you should update your implementation's `trans` and `transChoice` methods to `get` and `choice`.

The `Lang::getFromJson` Method

Likelihood Of Impact: Medium

The `Lang::get` and `Lang::getFromJson` methods have been consolidated. Calls to the `Lang::getFromJson` method should be updated to call `Lang::get`.



You should run the `php artisan view:clear` Artisan command to avoid Blade errors related to the removal of `Lang::transChoice`, `Lang::trans`, and `Lang::getFromJson`.

Mail

Mandrill & SparkPost Drivers Removed

Likelihood Of Impact: Low

The `mandrill` and `sparkpost` mail drivers have been removed. If you would like to continue using either of these drivers, we encourage you to adopt a community maintained package of your choice that provides the driver.

Notifications

Nexmo Routing Removed

Likelihood Of Impact: Low

A lingering part of the Nexmo notification channel was removed from the core of the framework. If you're relying on routing Nexmo notifications you should manually implement the `routeNotificationForNexmo` method on your notifiable entity [as described in the documentation](#).

Password Reset

Password Validation

Likelihood Of Impact: Low

The `PasswordBroker` no longer restricts or validates passwords. Password validation was already being handled by the `ResetPasswordController` class, making the broker's validations redundant and impossible to customize. If you are manually using the `PasswordBroker` (or `Password` facade) outside of the built-in `ResetPasswordController`, you should validate all passwords before passing them to the broker.

Queues

Queue Retry Limit

Likelihood Of Impact: Medium

In previous releases of Laravel, the `php artisan queue:work` command would retry jobs indefinitely. Beginning with Laravel 6.0, this command will now try a job one time by default. If you would like to force jobs to be tried indefinitely, you may pass `0` to the `--tries` option:

```
php artisan queue:work --tries=0
```

In addition, please ensure your application's database contains a `failed_jobs` table. You can generate a migration for this table using the `queue:failed-table` Artisan command:

```
php artisan queue:failed-table
```

Requests

The `Input` Facade

Likelihood Of Impact: Medium

The `Input` facade, which was primarily a duplicate of the `Request` facade, has been removed. If you are using the `Input::get` method, you should now call the `Request::input` method. All other calls to the `Input` facade may simply be updated to use the `Request` facade.

Scheduling

The `between` Method

Likelihood Of Impact: Low

In previous releases of Laravel, the scheduler's `between` method exhibited confusing behavior across date boundaries. For example:

```
$schedule->command('list')->between('23:00', '4:00');
```

For most users, the expected behavior of this method would be to run the `list` command every minute for all minutes between 23:00 and 4:00. However, in previous releases of Laravel, the scheduler ran the `list` command every minute between 4:00 and 23:00, essentially swapping the time thresholds. In Laravel 6.0, this behavior has been corrected.

Storage

Rackspace Storage Driver Removed

Likelihood Of Impact: Low

The `rackspace` storage driver has been removed. If you would like to continue using Rackspace as a storage provider, we encourage you to adopt a community maintained package of your choice that provides this driver.

URL Generation

Route URL Generation & Extra Parameters

In previous releases of Laravel, passing associative array parameters to the `route` helper or `URL::route` method would occasionally use these parameters as URI values when generating URLs for routes, even if the parameter value had no matching key within the route path. Beginning in Laravel 6.0, these values will be attached to the query string instead. For example, consider the following route:

```
Route::get('/profile/{location}', function ($location = null) {  
    //  
})
```

```
}->name('profile');

// Laravel 5.8: http://example.com/profile/active
echo route('profile', ['status' => 'active']);

// Laravel 6.0: http://example.com/profile?status=active
echo route('profile', ['status' => 'active']);
```

The `action` helper and `URL::action` method are also affected by this change:

```
Route::get('/profile/{id}', 'ProfileController@show');

// Laravel 5.8: http://example.com/profile/1
echo action('ProfileController@show', ['profile' => 1]);

// Laravel 6.0: http://example.com/profile?profile=1
echo action('ProfileController@show', ['profile' => 1]);
```

Validation

FormRequest `validationData` Method

Likelihood Of Impact: Low

The form request's `validationData` method was changed from `protected` to `public`. If you are overriding this method in your implementation, you should update the visibility to `public`.

Miscellaneous

We also encourage you to view the changes in the [laravel/laravel GitHub repository](#). While many of these changes are not required, you may wish to keep these files in sync with your application. Some of these changes will be covered in this upgrade guide, but others, such as changes to configuration files or comments, will not be. You can easily view the changes with the [GitHub comparison tool](#) and choose which updates are important to you.

Become a Laravel Partner

Laravel Partners are elite shops providing top-notch Laravel development and consulting. Each of our partners can help you craft a beautiful, well-architected project.

[Our Partners](#)

Laravel

Highlights

[Release Notes](#)
[Getting Started](#)
[Routing](#)
[Blade Templates](#)
[Authentication](#)
[Authorization](#)
[Artisan Console](#)
[Database](#)
[Eloquent ORM](#)
[Testing](#)

Resources

[Laracasts](#)
[Laravel News](#)
[Laracon](#)
[Laracon EU](#)
[Laracon AU](#)
[Jobs](#)
[Certification](#)
[Forums](#)

Partners

[Vehid](#)
[Tighten Co.](#)
[Kirschbaum](#)
[Byte 5](#)
[64Robots](#)
[Cubet](#)
[DevSquad](#)
[Ideil](#)
[Cyber-Duck](#)
[ABOUT YOU](#)
[Become A Partner](#)

Ecosystem

[Vapor](#)
[Forge](#)
[Envoyer](#)
[Horizon](#)
[Lumen](#)
[Nova](#)
[Echo](#)
[Valet](#)
[Mix](#)
[Spark](#)
[Cashier](#)
[Homestead](#)
[Dusk](#)
[Passport](#)
[Scout](#)
[Socialite](#)

Laravel is a web application framework with expressive, elegant syntax. We believe development must be an enjoyable and creative experience to be truly fulfilling. Laravel attempts to take the pain out of development by easing common tasks used in most web projects.

Laravel is a Trademark of Taylor Otwell.
Copyright © 2011-2019 Laravel LLC.

