

Upgrade Guide

- [Upgrading To 12.0 From 11.x](#)

High Impact Changes

- [Updating Dependencies](#)
- [Updating the Laravel Installer](#)

Medium Impact Changes

- [Models and UUIDv7](#)

Low Impact Changes

- [Carbon 3](#)
- [Concurrency Result Index Mapping](#)
- [Container Class Dependency Resolution](#)
- [Image Validation Now Excludes SVGs](#)
- [Multi-Schema Database Inspecting](#)
- [Nested Array Request Merging](#)

Upgrading To 12.0 From 11.x

Estimated Upgrade Time: 5 Minutes

[!NOTE] 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. Want to save time? You can use [Laravel Shift](#) to help automate your application upgrades.

Updating Dependencies

Likelihood Of Impact: High

You should update the following dependencies in your application's `composer.json` file:

- `laravel/framework` to `^12.0`
- `phpunit/phpunit` to `^11.0`
- `pestphp/pest` to `^3.0`

Carbon 3

Likelihood Of Impact: Low

Support for [Carbon 2.x](#) has been removed. All Laravel 12 applications now require [Carbon 3.x](#).

Updating the Laravel Installer

If you are using the Laravel installer CLI tool to create new Laravel applications, you should update your installer installation to be compatible with Laravel 12.x and the [new Laravel starter kits](#). If you installed the Laravel installer via `composer global require`, you may update the installer using `composer global update`:

```
composer global update laravel/installer
```

If you originally installed PHP and Laravel via `php.new`, you may simply re-run the `php.new` installation commands for your operating system to install the latest version of PHP and the Laravel installer:

```
/bin/bash -c "$(curl -fsSL https://php.new/install/mac/8.4)"
```

```
# Run as administrator...
Set-ExecutionPolicy Bypass -Scope Process -Force;
[System.Net.ServicePointManager]::SecurityProtocol =
[System.Net.ServicePointManager]::SecurityProtocol -bor 3072; iex ((New-Object
System.Net.WebClient).DownloadString('https://php.new/install/windows/8.4'))
```

```
/bin/bash -c "$(curl -fsSL https://php.new/install/linux/8.4)"
```

Or, if you are using [Laravel Herd's](#) bundled copy of the Laravel installer, you should update your Herd installation to the latest release.

Authentication

Updated `DatabaseTokenRepository` Constructor Signature

Likelihood Of Impact: Very Low

The constructor of the `Illuminate\Auth\Passwords\DatabaseTokenRepository` class now expects the `$expires` parameter to be given in seconds, rather than minutes.

Concurrency

Concurrency Result Index Mapping

Likelihood Of Impact: Low

When invoking the `Concurrency::run` method with an associative array, the results of the concurrent operations are now returned with their associated keys:

```
$result = Concurrency::run([
    'task-1' => fn () => 1 + 1,
    'task-2' => fn () => 2 + 2,
]);

// ['task-1' => 2, 'task-2' => 4]
```

Container

Container Class Dependency Resolution

Likelihood Of Impact: Low

The dependency injection container now respects the default value of class properties when resolving a class instance. If you were previously relying on the container to resolve a class instance without the default value, you may need to adjust your application to account for this new behavior:

```
class Example
{
    public function __construct(public ?Carbon $date = null) {}
}

$example = resolve(Example::class);

// <= 11.x
$example->date instanceof Carbon;

// >= 12.x
$example->date === null;
```

Database

Multi-Schema Database Inspecting

Likelihood Of Impact: Low

The `Schema::getTables()`, `Schema::getViews()`, and `Schema::getTypes()` methods now include the results from all schemas by default. You may pass the `schema` argument to retrieve the result for the given schema only:

```
// All tables on all schemas...
$tables = Schema::getTables();

// All tables on the 'main' schema...
$table = Schema::getTables(schema: 'main');

// All tables on the 'main' and 'blog' schemas...
$table = Schema::getTables(schema: ['main', 'blog']);
```

The `Schema::getTableListing()` method now returns schema-qualified table names by default. You may pass the `schemaQualified` argument to change the behavior as desired:

```
$tables = Schema::getTableListing();
// ['main.migrations', 'main.users', 'blog.posts']

$table = Schema::getTableListing(schema: 'main');
// ['main.migrations', 'main.users']

$table = Schema::getTableListing(schema: 'main', schemaQualified: false);
// ['migrations', 'users']
```

The `db:table` and `db:show` commands now output the results of all schemas on MySQL, MariaDB, and SQLite, just like PostgreSQL and SQL Server.

Eloquent

Models and UUIDv7

Likelihood Of Impact: Medium

The `HasUuids` trait now returns UUIDs that are compatible with version 7 of the UUID spec (ordered UUIDs). If you would like to continue using ordered UUIDv4 strings for your model's IDs, you should now use the `HasVersion4Uuids` trait:

```
use Illuminate\Database\Eloquent\Concerns\HasUuids; // [tl! remove]
use Illuminate\Database\Eloquent\Concerns\HasVersion4Uuids as HasUuids; // [tl! add]
```

The `HasVersion7Uuids` trait has been removed. If you were previously using this trait, you should use the `HasUuids` trait instead, which now provides the same behavior.

Requests

Nested Array Request Merging

Likelihood Of Impact: Low

The `$request->mergeIfMissing()` method now allows merging nested array data using "dot" notation. If you were previously relying on this method to create a top-level array key containing the "dot" notation version of the key, you may need to adjust your application to account for this new behavior:

```
$request->mergeIfMissing([
    'user.last_name' => 'Otwell',
]);
```

Validation

Image Validation Now Excludes SVGs

The `image` validation rule no longer allows SVG images by default. If you would like to allow SVGs when using the `image` rule, you must explicitly allow them:

```
use Illuminate\Validation\Rules\File;

'photo' => 'required|image:allow_svg'

// Or...
'photo' => ['required', File::image(allowSvg: true)],
```

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.