

Changelog

New updates and improvements to Laravel's open source products.

February 2026

Laravel Framework 12.x

Add Support for `temporaryUploadUrl` to the `local` Filesystem

Pull request by [@mnapoli](#)

You can now generate [temporary upload URLs](#) even when you're using the `local` driver, making it much easier to build consistent upload flows across local, staging, and production environments.

```
1 use Illuminate\Support\Facades\Storage;
2
3 $url = Storage::disk('local')->temporaryUploadUrl(
4     'uploads/incoming/video.mp4',
5     now()->addMinutes(10),
6     ['Content-Type' => 'video/mp4']
7 );
```

[12.x] Add `makeMany` Method to Factory

Pull request by [@jackbayliss](#)

`makeMany()` gives you a clean, expressive way to generate multiple in-memory model instances without persisting anything. It's ideal for tests, previews, and shaping data before a write.

```
1 $users = User::factory()->makeMany(3);
2
3 $payload = $users->map->only(['name', 'email']);
```

[12.x] Allow Closures for Values in `firstOrCreate` and `createOrFirst`

Pull request by [@gcavanunez](#)

You can now lazily compute attribute values only when a record is actually being created. That means fewer unnecessary queries and less wasted work when the model already exists.

```
1 $location = Location::firstOrCreate(
2     ['address' => $address],
3     fn () => ['coordinates' => Geocoder::resolve($address)],
```

```
4     );
```

[12.x] Support afterSending Method on Notification

Pull request by [@gdebrauwer](#)

`afterSending()` lets you hook into the lifecycle after a [notification](#) is dispatched, which is perfect for audit logging, metrics, or cleanup work without cluttering your notification channels.

```
1  class BookingNotification extends Notification
2  {
3      public function __construct(
4          public Booking $booking;
5      ) {
6          //
7      }
8
9      public function via()
10     {
11         return ['mail'];
12     }
13
14     public function toMail()
15     {
16         // ...
17     }
18
19     public function afterSending($notifiable, $channel, $response)
20     {
21         $this->booking->update(['notified_at' => now()]);
22     }
23 }
```

[12.x] Add whenFails and whenPasses Methods on Validator

Pull request by [@gdebrauwer](#)

These helpers are particularly useful when you use the validator to validate data outside the request cycle where the error would not get picked up automatically by the HTTP handler.

```
1  Validator::make(
2      ['file' => $file],
3      ['file' => 'required|image|dimensions:min_width=100,min_height=200']
4  )->whenFails(function () {
5      throw new InvalidArgumentException('Provided file is invalid');
6  });
```

[12.x] Add `withoutHeader` Method to Response

Pull request by [@miladev95](#)

Removing headers is now straightforward with `withoutHeader()`, which is handy when you need to strip inherited headers (like caching or debug headers) before returning a [response](#).

```
1 return response($content)→withoutHeader('X-Debug');
```

[12.x] Exclude Decorative ASCII Art SVG from Exception Page in Non-Browser Contexts

Pull request by [@serhiilabs](#)

Console and non-browser contexts no longer receive the decorative ASCII SVG payload, keeping error output cleaner in logs, CI, agent, and API responses. Furthermore...

[12.x] Use JS to Create the Laravel ASCII SVG Logo on the Fly

Pull request by [@markjaquith](#)

The exception page ASCII SVG is now generated on the fly with JavaScript, keeping output leaner and better suited to different rendering contexts.

[12.x] Add Typed Getters on Cache

Pull request by [@ahinkle](#)

Typed cache getters reduce repetitive casting and make intent obvious when reading cached values, helping you avoid subtle bugs from unexpected types.

```
1 $count = Cache::integer('downloads.count', 0);
2 $enabled = Cache::boolean('features.new_checkout', false);
```

[12.x] Add `InteractsWithData::clamp()`

Pull request by [@cosmastech](#)

A `clamp()` method has been added to the `InteractsWithData` trait, giving you a tidy way to bound numeric input values, which is great for pagination, limits, sliders, and any user-controlled numbers.

```
1 // Instead of:
2 /** @var int<PHP_INT_MIN, PHP_INT_MAX> $perPage */
3 $perPage = request()→integer('per_page', 50);
4
5 // You can now do:
```

```
5 // You can now do:
6 /** @var int<1, 100> $perPage */
7 $perPage = request()→clamp('per_page', 1, 100, 50);
```

AI SDK

Make Provider Default Models Configurable

Pull request by [@pfrug](#)

The default models used for text, images, audio, transcription, and embeddings may are now configurable in your application's `config/ai.php` file. This gives you granular control over the exact models you'd like to use if you don't want to rely on the package defaults.

```
1 return [
2     'models' => [
3         'text' => [
4             'default' => 'claude-sonnet-4-6',
5             'cheapest' => 'claude-haiku-4-5-20251001',
6             'smartest' => 'claude-opus-4-6',
7         ],
8     ],
9 ],
```

Add Support for Timeouts in Transcription

Pull request by [@NietThijmen](#)

Transcription now supports timeouts, giving you better control in production workloads and preventing long-running requests from tying up workers.

```
1 $transcript = Transcription::fromPath('./podcast.mp3')→timeout(240)→generate(La
```

New Providers Added

Azure OpenAI

Pull request by [@imYashGupta](#)

Mistral AI

Pull request by [@jackbayliss](#)

Voyage AI

Pull request by [@jackbayliss](#)

DeepSeek

Pull request by [@ihxnnxs](#)

Boost

Added Support for Loading Guidelines and Skills Directly from Vendor Packages

Pull request by [@pushpak1300](#)

Guidelines and skills can now be loaded directly from vendor packages, ensuring that guidelines always match the package version you are using and will never go stale.

Add Support for Nightwatch MCP

Pull request by [@jessarcher](#)

Nightwatch MCP support improves the integration story for observability-driven workflows, so you can connect monitoring context to your tools more seamlessly.

Enhance Database-Schema Tool with Full Column Metadata

Pull request by [@alanost](#)

The database schema tool now exposes richer column metadata, which improves the quality of schema-aware tooling and reduces incorrect assumptions (types, defaults, nullability).

This is particularly helpful when generating code, validating migrations, or building assistants that reason about your database structure.

MCP

Change JSON Response Formatting to Compact by Default

Pull request by [@pushpak1300](#)

JSON responses are now compact by default, trimming payload size and making tool-to-tool communication faster and less noisy in logs.

Prompts

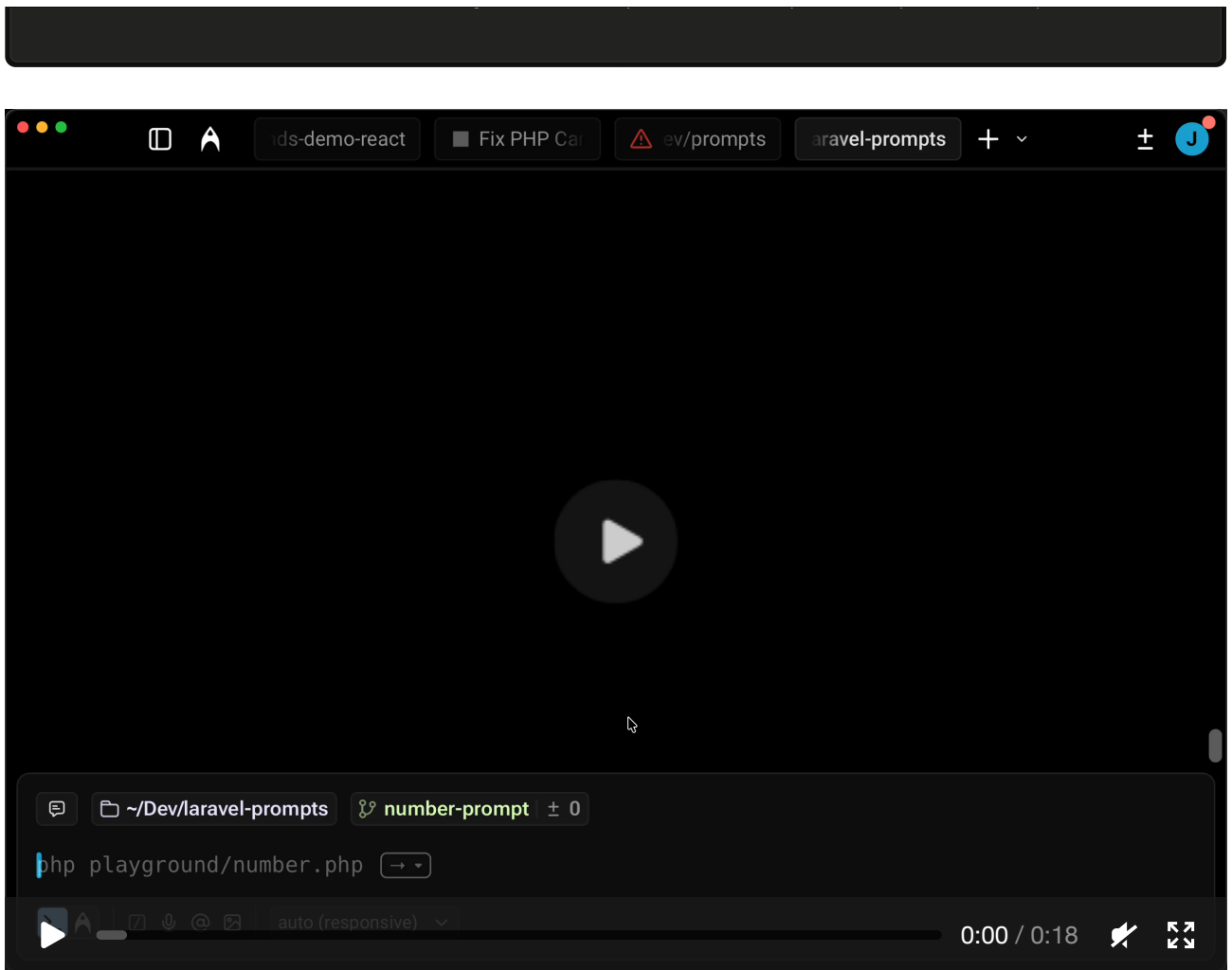
Add `number` Prompt

Pull request by [@joetannenbaum](#)

A new `number` [prompt](#) makes it easy to collect numeric input with validation baked in, perfect for CLI setup flows and interactive tooling. Features automatic clamping of the min/max values and adjusting the value via the up and down arrows.

```
1 $retries = number('How many retries?', default: 3, min: 0, max: 10);
```





Stream

Svelte Package

Pull request by [@joetannenbaum](#)

The new `@laravel/stream-svelte` package makes it trivial to [consume streaming output](#) from your server in your Svelte app. If you're shipping real-time UI, this helps you move from prototype to production faster.

Svelte Starter Kit

Pull request by [@joetannenbaum](#)

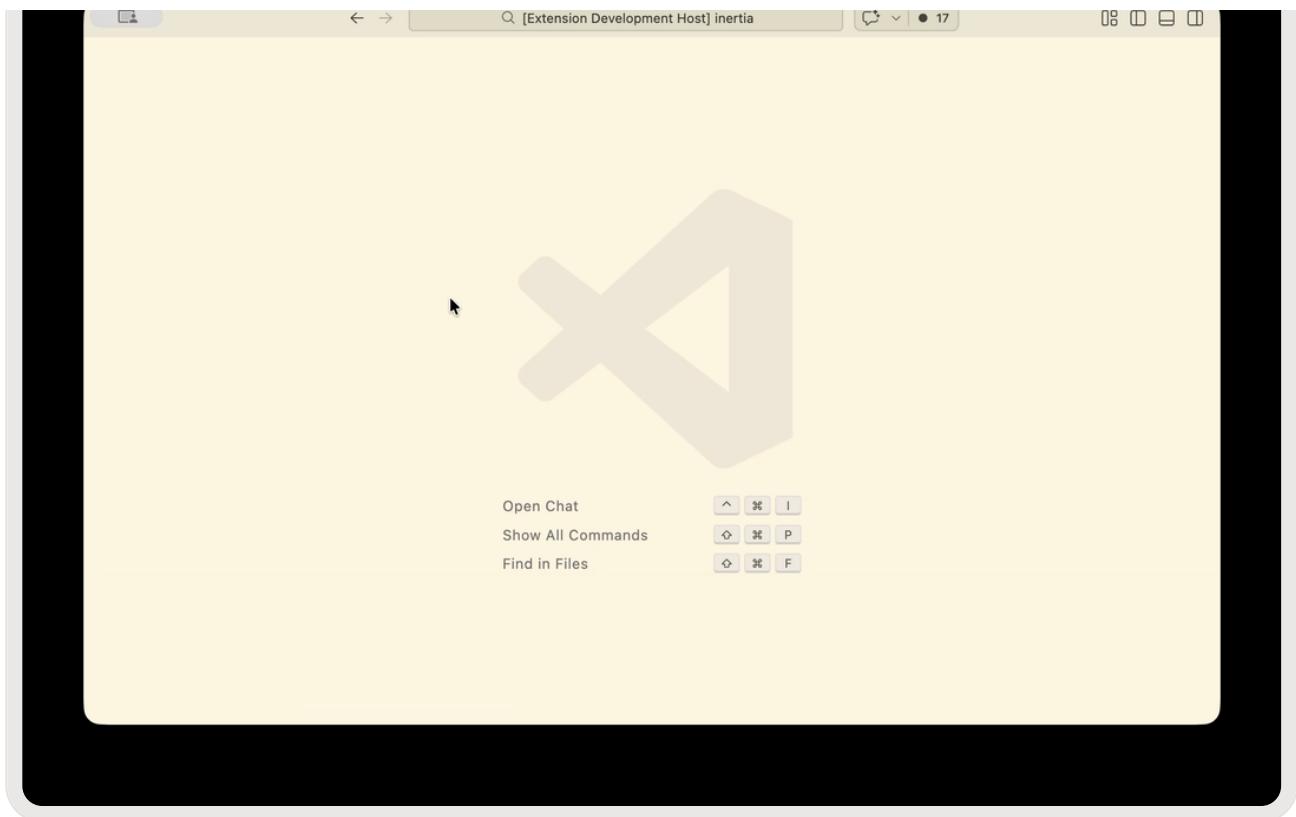
The new Svelte [starter kit](#) gets you up and running with Inertia, Svelte, and Laravel conventions quickly, so you can spend time on product code instead of scaffolding.

VS Code Extension

Test Runner Integration

Pull request by [@TitasGailius](#)

Integrating with VS Code's built-in test runner makes it easier to run and iterate on your suite from within your IDE, shortening the edit-test loop and helping you stay in flow.



Livewire Props

Pull request by [@TitasGailius](#)

New support for hovering over Livewire 3 and 4 props means that you can see the relevant type and docblock information directly from the PHP class, reducing context switching in your Livewire and Blade components.

```
resources/views/livewire/features/⚡ button/button.blade.php
```

```
<?php
```

```
string $variant = 'solid'
```

```
Follow link (cmd + click)
```



Laravel is the most productive way to
build, deploy, and monitor software.



© 2026 Laravel

[Legal](#)

[Status](#)

Products

[Cloud](#)

[Forge](#)

[Nightwatch](#)

[Vapor](#)

[Nova](#)

Packages

[Cashier](#)

[Dusk](#)

[Horizon](#)

[Octane](#)

[Scout](#)

[Pennant](#)

[Pint](#)

[Sail](#)

[Sanctum](#)

[Socialite](#)

[Telescope](#)

[Pulse](#)

[Reverb](#)

[Echo](#)

Resources

[Documentation](#)

[Starter Kits](#)

[Release Notes](#)

[Blog](#)

[News](#)

[Community](#)

[Larabelles](#)

[Learn](#)

[Jobs](#)

[Careers](#)

[Trust](#)

Partners

[Pixel](#)

[Steadfast Collective](#)

[Vehikl](#)

[Bacancy](#)

[Curotec](#)

[Jump24](#)

[Threadable](#)

[64 Robots](#)

[CACI Limited](#)

[Tighten](#)

[See All](#)

Laravel