

The story behind our open source efforts

5th January 2018

This post was first published on 24daysindecember.net

Chances are that you've never heard of my company [Spatie](#). We specialize in creating Laravel applications for our clients. Our team is rather small: we consist of only 6 developers and one manager. On first glance we are just a web agency like there are so many others. But there's one thing that sets our company apart: we have an open source first policy. We try to create and contribute to open source as much as possible.

Currently we have around 170 open source repositories on GitHub. Our packages have been downloaded nearly 8 million times. They are being downloaded 800 000 times each month. Git awards is a site that ranks GitHub organisations based on the amount of stars their repos have. This past year we managed to take [the #3 spot on Git Awards' worldwide PHP developers list](#). And yeah, I know that the amount of stars is not an important metric, but it's still a nice achievement for a smallish team.

This all did not happen overnight. Read on to learn the story behind our open source efforts.

The first package

When I got started to work for Spatie, somewhere around 2007, I dared to write my own framework. The projects I was working on at that time were very small. Submitting a simple form was the most advanced thing these sites should do.

After a few years our projects got a bit more advanced. I realised that writing and maintaining my own framework wasn't going to cut it anymore. I heard about this fancy thing called Zend Framework 1 and starting using that.

Let's skip ahead a few more years. In 2012 and 2013 I felt that the PHP ecosystem was pretty stale and I considered switching to Ruby and the Rails framework. But then I bumped into Laravel 4.0. I was immediately impressed by the expressive syntax and the big focus on developer happiness. I used it for a couple of projects and the more I learned about it, the more I loved using it.

At that time a guy called [Jeffrey Way](#) started his next project [Laracasts](#): a video tutorial site dedicated to Laravel. Realising that his videos could speed up my learning process immensely I immediately bought a lifetime subscription.

On the 4th of March in 2014 Jeffrey published a video titled “[Continuous Integration With Travis](#)“. My mind was blown. [Travis](#) and the integration with GitHub looked so cool that I wanted to use it. A couple of months before he made the Travis video Jeffrey made a miniseries [on package development](#). Those videos sparked the thought in my mind that I could create a package of my own.

In a project at the time we needed to automatically create screenshots of a website. With the things I learned in the Laracasts-videos I started working on my first package [Browsershot](#). In essence it was just a simple wrapper around [PhantomJS](#) ([the current version](#) uses Headless Chrome and [Puppeteer](#)).

It was pretty exciting working in the open. I was thrilled every time the download counter got up and was incredibly happy that other people started [writing about it](#).

More packages! #

In the Zend Framework days I made used a custom-built application template, called Blender, to kickstart all client projects. It was a sort of mini-cms. Beside the traditional CMS-functions Blender could do a lot of cool things: pull in data from Analytics to show graphs about the usage of the site, handle uploaded files, subscribe users to a MailChimp list, ...

To power new projects I needed a Laravel powered version of Blender and started creating it from scratch. I soon realised that some of the things I was porting to Laravel could be useful for other developers as well and decided to export these functionalities to external packages.

In quick succession packages like [laravel-newsletter](#), [laravel-medialibrary](#) and [laravel-analytics](#) were made.

I started to enjoy package development more and more. The positive feedback that I got early on from package users was very encouraging. I always considered coding purely as work, but because it was so much fun creating code that others could use as well I started coding in my free time as well.

Besides working on packages I also polished the aforementioned [application template called Blender](#) and open sourced that too.

Currently every new package that we create gets born inside a client project. In almost every project we find some functionality that can be extracted to it's own package.

The benefits of creating packages #

There are many benefits of creating packages that makes time working on open source code well spent.

A lot can be learned while creating a package. Each package needs to be carefully crafted. It provides a clear and understandable syntax. Like Laravel, I want that all our packages put a great emphasis on developer happiness. They should be easy to use. They should have great documentation. They should have clear tests. Thinking about how other people will use your code will make you a better developer.

The issues reported and the PRs submitted by the users of our packages provided another chance of learning. People can point to mistakes that are in our packages and can propose interesting new features that we didn't think about.

Sometimes, as a package gains some traction, you get quality code for free. Our [laravel-fractal](#) package is a developer friendly wrapper around [The League's Fractal package](#). I coded up the basic functionality to I needed myself and tagged that 1.0.0. In the next weeks I almost daily got a pull request adding another great feature to the package. And now it supports almost everything League's Fractal can do. I think 90% of the code of that package was written by the community.

Of course we're also [dogfooding ourselves](#). Our own packages get used on most projects. If we discover a bug in a package used in a project we can very quickly fix that and distribute that fix through the power of Composer to our other projects.

There are also commercial benefits. The past year we've landed some cool projects because of our open source work. In most cases these new clients were already using our stuff. Even though attracting clients isn't the primary goal of our open source efforts, it sure is a nice side effect.

The positive feedback from package users also gave me enough confidence to start [a blog](#) and start speaking publicly. The past years I've spoken at several local users groups (and I'm co-organizing [the one in my home town](#)) and a lot of conferences world wide. The past few years I had the pleasure of giving talks in Washington, Amsterdam, Tokyo, New Delhi, New York, Istanbul and of course my hometown Antwerp. At [all those occasions](#) I got to meet a lot of awesome fellow developers, including [Andreas](#), who runs <https://24daysindecember.net>. That all probably wouldn't have happened if I stayed in my little bubble. Working on open source helped getting to connect to the community. That sounds a bit corny, but that doesn't make it less true.

Making time to work on open source #

People often ask if it isn't very time-consuming to create packages. The truth is that it indeed takes a lot of time. Creating the code for the package itself, writing tests, writing documentation and getting the word out all takes a fair amount of time.

And that's just the start. When a first stable release gets tagged the work is not over. Maintaining a package, responding to issues, reviewing PR's takes a lot of time (and dedication) too. At the time of this writing we've responded to [over 3 000 issues](#) and reviewed [more a lot of PRs](#).

We do long-term planning at Spatie, but we also have a weekly short-term planning meeting. When scheduling out to the coming week we only plan four days. So we have one work day we can be a bit flexible with. Do not imagine that day as like a fixed day, that time is mostly spread out in that week.

In that "fifth" day everybody in our company is allowed to work on open source stuff or their own projects (which, if possible, will be opensourced as well). For our employees this has a nice benefit that they can keep things interesting by switching between working on client projects and open source stuff.

Personally I do enjoy working on packages in my free time as well. So a few times a week I spend some time in the evening to continue improving the released packages and create new stuff.

Meet the team #

Because I blog, tweet and try to speak at conferences a lot I've become the public face of the company's open source efforts. But you should know that I'm not doing it all by myself. I've got a great team behind me that does a lot of the work.

Since the start of Spatie I was the sole back-end developer at our company. That changed mid 2015 when [Seb](#) joined our team. He loves working on the more lower level PHP and JavaScript stuff. He's the primary author of the [regex](#), [schema-org](#), [phpunit-snapshot-assertions](#) and many many more packages. And our [dashboard](#) and [JavaScript](#) code wouldn't be as good if it weren't for him. Be sure to bookmark [his blog](#).

Last year we had a great intern, [Alex](#), who has meanwhile joined our team. He's helping out maintaining [laravel-permission](#), which has gotten [quite](#)

[popular](#). He's currently doing some kick-ass work on some non-released packages / projects that I can't talk about yet.

A couple of months ago we hired [Harish](#), our first remote employee. He's helping out with issues across all our repos and laid the foundation of [our dns package](#).

[Brent](#), our latest hire, likes to focus on performance. His passion surely reflects on the opensource work he's been doing. He created the [laravel-binary-uuid](#) package, he pushed me (in a friendly way) to add [support for responsive images](#) to our medialibrary. Currently he's busy creating [a very cool async package](#). He also has [a cool blog](#).

And last but not least there's [Willem](#), who is the founder of Spatie. He's responsible for the good looks of [our dashboard](#), our [dns lookup service](#), the various [demo pages of our packages](#), the copy of [our Patreon page](#) and much much more.

I'm really happy that every member of our team enjoys working on open source. Hopefully they all will keep working for Spatie in the foreseeable future. I do hope that if somebody would leave us for greener pastures, that they'll bring the open source spirit to their new endeavours.

Top 10 of most popular packages

When taking amount of downloads into consideration these are our 10 most popular packages:

1. [laravel-backup](#)
2. [laravel-fractal](#)
3. [db-dumper](#)
4. [laravel-permission](#)
5. [pdf-to-image](#)
6. [laravel-medialibrary](#)
7. [fractalistic](#)
8. [laravel-newsletter](#)
9. [laravel-glide](#)

10. [laravel-tail](#)

Some of these packages, like [db-dumper](#), probably get a big download boost because there are required by other packages.

Not so popular packages

There are some packages, that we ourselves use in every single project, that I expected to be more popular. I hope by mentioning them here they'll get a little attention. If you decide to give the ones mentioned below a shot, I hope that you'll enjoy using them as much as we do.

The first one in this category is [laravel-uptime-monitor](#). This package can check the uptime of multiple sites. It will notify you when your site is down (and when it comes back up). You can also be notified a few days before an SSL certificate on one of your sites expires. Under the hood, the package leverages Laravel native notifications, so it's easy to use Slack, Telegram or your preferred notification provider.

Secondly, there's [laravel-translatable](#). This one can make your Eloquent models hold translations. Unlike [Dimitris Savvopoulos' excellent translatable package](#), ours does not need separate translation tables to store translations. Our package stores the translations as json in a column of the table. It's very easy to work with, there are no extra queries needed to fetch translations. We got the idea to store translations as json in a column from [Mohamed Said](#) (who now enjoys fame as Laravel employee #1 ?).

Even though you saw [laravel-medialibrary](#) in the top 10 above, I think this one deserves to be a few places higher. The package can associate all kind of media files (images, PDFs, ...) with Eloquent models. It can generate thumbnails, and can store files on external filesystem and generate URLs to the files. It's very easy to work with and it has [excellent documentation](#). Our team is already hard at work in creating v7 which will add support for zipping assets on the fly, responsive images, client side components and much more.

How can you show appreciation for our efforts

Maybe I should have mentioned it before, but our packages are not entirely free. There's a special license on them: [Postcardware](#). If any of our stuff makes it into your production environment, we'd hight appreciate you sending us a postcard. This is our address:

Spatie bvba
Samberstraat 69D
2060 Antwerp
Belgium

Every week the postman delivers some postcards to us. For my team and I it's nice to know that our stuff is being used all over the globe. We've gotten postcards from all continents. We even got one from [Vanuatu](#) which, for us, is litteraly on the other side of the globe. You can take a look at our postcard collection as well. They all are [published](#) on our website.

Another way you can show your appreciation is by becoming one of our patrons. Like mentioned on [our Patreon page](#) your donation will help us to dedicate even more time to maintain and create new open source code.

You can also give us a pat on the back by starring one of [our repos on GitHub](#).

The future

We don't have a fixed roadmap for new packages. It depends a bit on the client projects we're handling. If there is something there that can be solved in a generic way, we'll create a package.

There's been a time when I thought that, because we'd already solved a lot of common problems, the package output of our team was going to diminish a bit. But it's quite the opposite. Almost every member of our team has some good package ideas.

One of the bigger things we're working on now as a team is the

aforementioned new upcoming version of the medialibrary. You can expect that one to drop somewhere Q1 2018.

In closing #

Creating and working on open source packages continues to be a truly fun experience. On a technical level I've learned lots of things that I wouldn't have learned when working solely on client projects. Like you've probably read between the lines, it also brings me great joy that my co-workers enjoy and appreciate the time given to work on our open source body of work.

I hope you've enjoyed the story behind our open source efforts. The big conclusion is that there are only winners in this story. If you are in a position in your company to advocate or to actively work on open source, I highly recommend you do so.

If you haven't used our packages before, take a look at [the list on our company website](#). Probably we've made some things that could be useful to you.