

API PLATFORM CON 2024 - LILLE, FRANCE & ONLINE



# API PLATFORM 4

The API-first framework

# About Me

---



**Kévin Dunglas**

 [dunglas@mastodon.social](mailto:dunglas@mastodon.social)

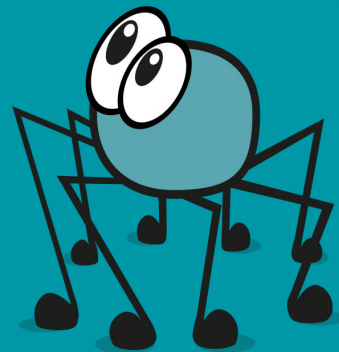
 [@dunglas](https://twitter.com/dunglas)

 [dunglas.dev](https://dunglas.dev)

- ✓ **API Platform** creator
- ✓ **Les-Tilleuls.coop** co-founder
- ✓ **Polyglot** developer

# API Platform Community

- ✓ **700+** conference attendees
- ✓ **8,442 stargazers** on GitHub
- ✓ **864** code and docs **contributors**
- ✓ **5,505** people on Slack



# Crossing the Atlantic





# Getting Started... with a New Stack

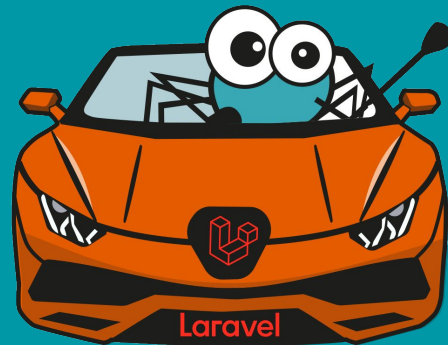
```
composer create-project laravel/laravel my-api  
cd my-api
```

```
# or open an existing Laravel project!
```



# Laravel In Numbers

- ✓ **Most popular web framework**  
(across all languages)
- ✓ **78,215 stargazers** on GitHub
- ✓ **3452 contributors**
- ✓ **40,432** users on Discord
- ✓ Just raised a **\$57 million Series A**  
from Accel



# Installing API Platform for Laravel 🎉



```
composer require api-platform/laravel
```

# Disclaimer: This Guy Wrote Most of the Code

“  
**API PLATFORM:  
THE PRAGMATIC  
FRAMEWORK**  
”

**ANTOINE  
BLUCHET**

**API PLATFORM  
PROJECT LEAD**

@s0yuka



**API PLATFORM  
CONFERENCE**  
SEP 19-20, 2024 | LILLE

**2024**

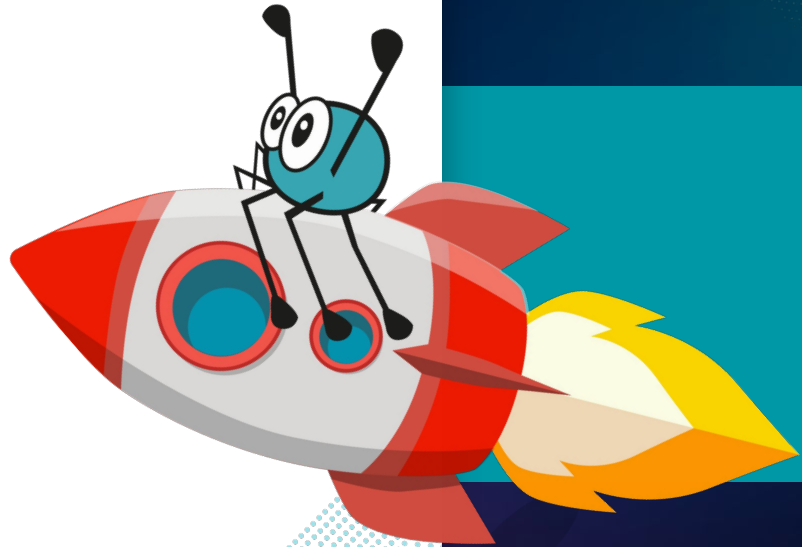
[API-PLATFORM.COM/CON](https://api-platform.com/con)

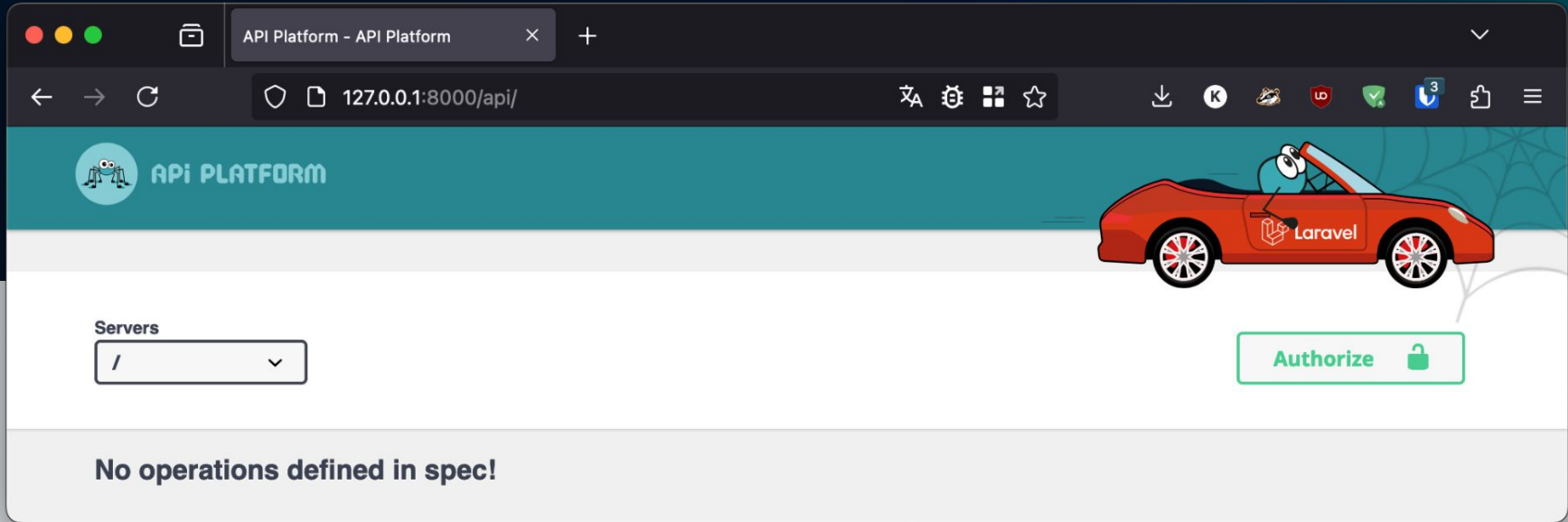
**API PLATFORM  
CONFERENCE**

# Start the Dev Web Server

---

```
php artisan serve
```

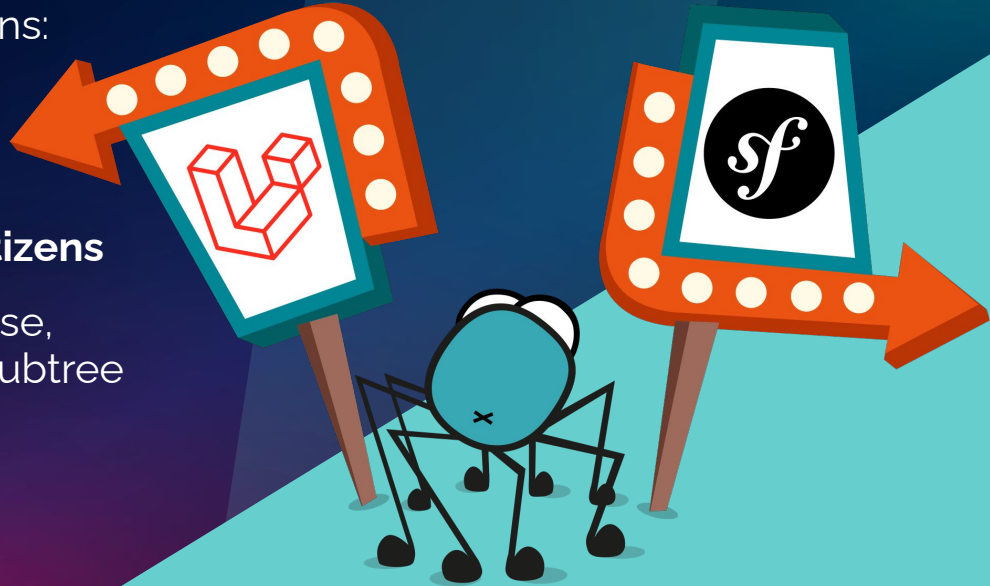




# Wasn't API Platform Symfony-specific?

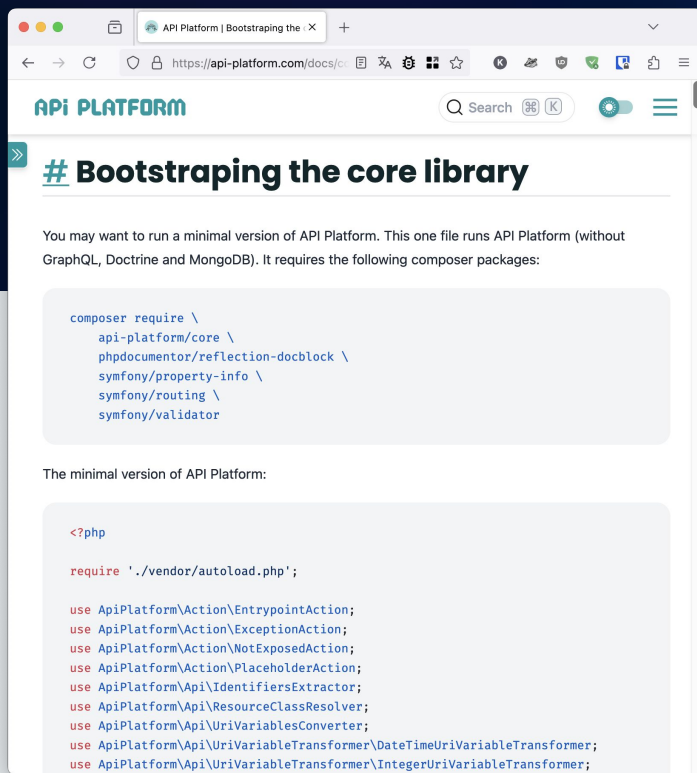
---

- ✓ Since **v2**: a standalone PHP library...
- ✓ ...with two official framework integrations:
  - a Symfony bundle from day one
  - a Laravel package as of today
- ✓ **Symfony and Laravel are 1st class citizens**
- ✓ Both run the **same code**: one code base, monolithic repository with read-only subtree split
- ✓ Most features (~90%) are **available for both frameworks**





# It's Even Possible To Use API Platform **Without** a Framework



The screenshot shows a web browser window displaying the API Platform documentation. The page title is "# Bootstrapping the core library". The content explains that a minimal version of API Platform can be run without GraphQL, Doctrine, and MongoDB, and lists the required Composer packages. Below this, there are two code blocks: one for the Composer require command and another for the minimal PHP bootstrap code.

API PLATFORM

## # Bootstrapping the core library

You may want to run a minimal version of API Platform. This one file runs API Platform (without GraphQL, Doctrine and MongoDB). It requires the following composer packages:

```
composer require \
  api-platform/core \
  phpdocumentor/reflection-docblock \
  symfony/property-info \
  symfony/routing \
  symfony/validator
```

The minimal version of API Platform:

```
<?php
require './vendor/autoload.php';

use ApiPlatform\Action\EntrypointAction;
use ApiPlatform\Action\ExceptionAction;
use ApiPlatform\Action\NotExposedAction;
use ApiPlatform\Action\PlaceholderAction;
use ApiPlatform\Api\IdentifiersExtractor;
use ApiPlatform\Api\ResourceClassResolver;
use ApiPlatform\Api\UriVariablesConverter;
use ApiPlatform\Api\UriVariableTransformer\DateTimeUriVariableTransformer;
use ApiPlatform\Api\UriVariableTransformer\IntegerUriVariableTransformer;
```

# What's Inside the Laravel Package?

- ✓ **Provider:** registers services provided by API Platform in Laravel Dependency Injection Container
- ✓ **Routing:** automatically registers and implements **REST** routes
- ✓ **Docs:** automatically generates **OpenAPI** (formerly known as Swagger) spec
- ✓ **GraphQL:** state-of-the-art GraphQL support implementing the Relay spec
- ✓ **Idiomatic** integrations with **the Laravel ecosystem:**
  - Eloquent, Validation, Gates, Policies, Octane, Broadcast, Pest...



# Create an Eloquent Model:

```
php artisan make:model Book
```

```
namespace App\Models;  
  
use Illuminate\Database\Eloquent\Model;  
  
class Book extends Model {}
```

# Eloquent

---

- ✓ Eloquent is the Laravel **ORM**
- ✓ It implements the Active Record pattern
- ✓ Table columns are **automatically exposed as magic model class properties**
- ✓ It can also be used as a standalone library, even in Symfony with `wouterj/eloquent-bundle`

# Eloquent (ctd)

---

- ✓ Eloquent supports SQLite, PostgreSQL, MySQL, MariaDB and SQL Server
- ✓ All you have to do is to create a table (books in our example) and add columns, **no code and no mapping required!**
- ✓ You can do it manually or use **migrations** (recommended)

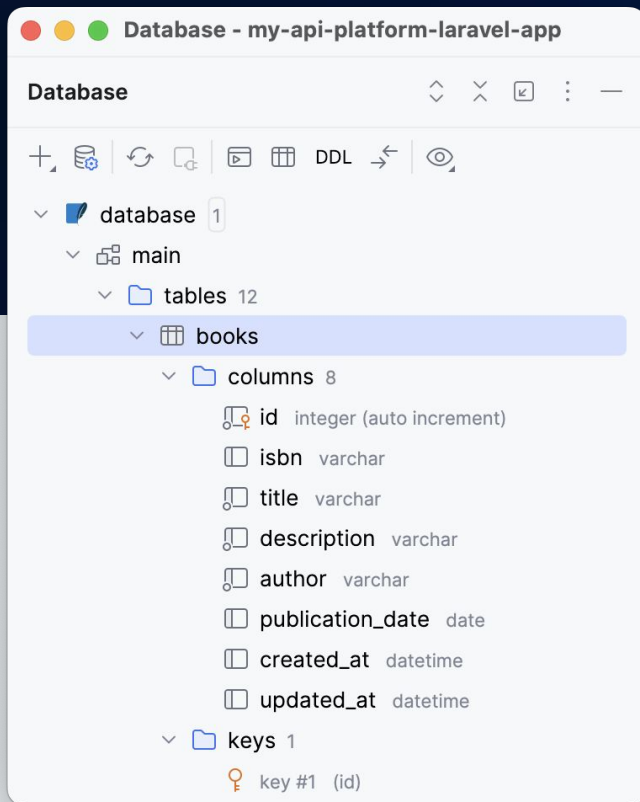
# Create a Migration:

php artisan make:migration

```
Schema::create('books', function (Blueprint $table) {  
    $table->id();  
  
    $table->string('isbn')->nullable();  
    $table->string('title');  
    $table->text('description');  
    $table->string('author');  
    $table->date('publication_date')->nullable();  
  
    $table->timestamps();  
});
```

# Run the Migration:

```
php artisan migrate
```





# Your State-of-the-Art API With One Attribute


```
namespace App\Models;

use ApiPlatform\Metadata\ApiResource;
use Illuminate\Database\Eloquent\Model;

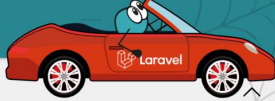
#[ApiResource]
class Book extends Model {}
```

API Platform - API Platform

127.0.0.1:8000/api/



API PLATFORM



## Book

GET	/api/books	Retrieves the collection of Book resources.	▼
POST	/api/books	Creates a Book resource.	▼
GET	/api/books/{id}	Retrieves a Book resource.	▼
DELETE	/api/books/{id}	Removes the Book resource.	▼
PATCH	/api/books/{id}	Updates the Book resource.	▼

### Schemas

Book > Expand all object

Book.jsonld ^ Collapse all object

```
@context > Expand all read-only (string | object)
@id read-only string
@type read-only string
id read-only integer
isbn string | null
title string
description string
author string
publicationDate string | null date-time
createdAt string | null date-time
updatedAt string | null date-time
```



Request URL

http://127.0.0.1:8000/api/books?page=1

Server response

Code Details

200

Response body

```
{
  "@context": "/api/contexts/Book",
  "@id": "/api/books",
  "@type": "Collection",
  "totalItems": 100,
  "member": [
    {
      "@id": "/api/books/1",
      "@type": "Book",
      "id": 1,
      "isbn": "9780535875083",
      "title": "Quia Vel Omnis Reprehenderit",
      "description": "Quo omnis sit dolorem omnis magni beatae et alias. Vero et unde earum et vel it. Dolor cumque modi saepe velit nulla dolore unde. Eum voluptate laboriosam optio optio nihil arc hitecto ea dolor.",
      "author": "Walker Weber",
      "publicationDate": "1989-02-26",
      "createdAt": "2024-09-16T19:28:17+00:00",
      "updatedAt": "2024-09-16T19:28:17+00:00"
    },
    {
      "@id": "/api/books/2",
      "@type": "Book",
      "id": 2,
      "isbn": "9789552730047"
```

 Download

# Under the Hood

---

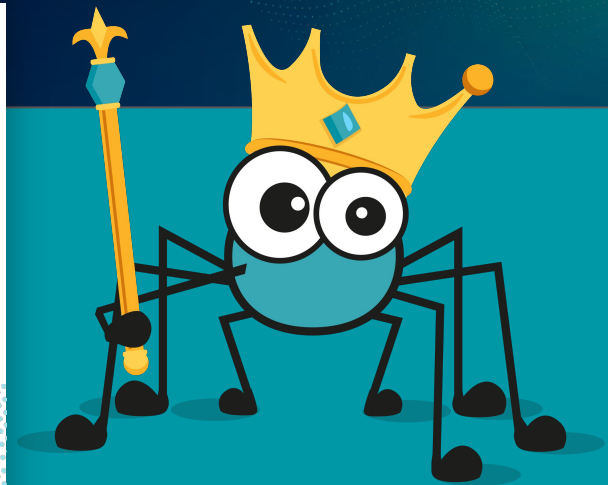
- ✓ API Platform introspects the database (column names, types, constraints...) to populate **metadata**
- ✓ Serialization, OpenAPI and Hydra docs... are generated from these metadata
- ✓ The same attributes as with the Symfony variant are used
- ✓ Everything is **entirely configurable and extensible**



# Unrivalled Compliance With Open Standards

By default, API Platform:

- ✓ exposes a true **REST** API (HATEOAS/hypermedia)
- ✓ follows standards as closely as possible:
  - W3C's **JSON-LD**/RDF with Schema.org, Hydra (interoperability, AI, big data)
  - IETF's RFC 7807 (API errors), 7386 (JSON Merge Patch), 8594 (endpoint deprecations), etc
- ✓ embraces **OWASP** REST security best practices



# Other Formats:

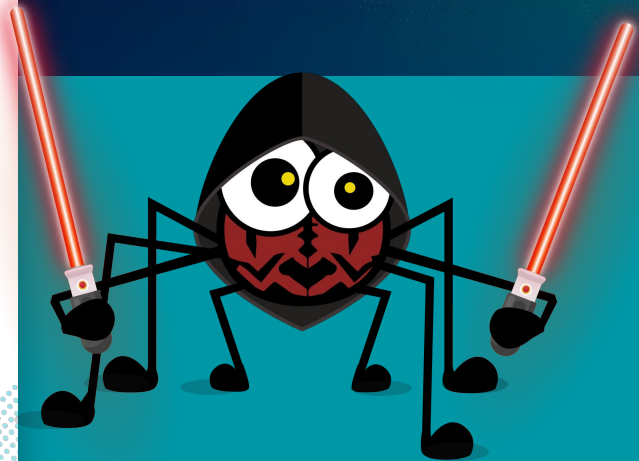
JSON:API, HAL and CSV  
(YAML is planned, PR welcome!)

```
// config/api-platform.php
return [
    'formats' => [
        'jsonld' => ['application/ld+json'],
        'jsonapi' => ['application/vnd.api+json'],
        'jsonhal' => ['application/hal+json'],
        'csv' => ['text/csv'],
    ],
    'docs_formats' => [
        'jsonld' => ['application/ld+json'],
        'jsonapi' => ['application/vnd.api+json'],
        'jsonopenapi' => ['application/vnd.openapi+json'],
        'html' => ['text/html'],
    ],
    // ...
];
```

# GraphQL? Challenge Accepted!

composer require api-platform/graphql

```
// config/api-platform.php  
'graphql' => [  
    'enabled' => true,  
// ...
```





API Platform - API Platform

127.0.0.1:8000/api/graphql?query={%0A%20 books(first%3A 3) {%0A%20%20%20 edges {%0A%20

## Docs

A GraphQL schema provides a root type for each kind of operation.

Root Types

query: Query

mutation: Mutation

All Schema Types

- ID
- Node
- Int
- String
- BookCursorConnection
- BookEdge
- Book
- BookPageInfo
- Boolean
- updateBookInput
- updateBookPayload
- deleteBookInput
- deleteBookPayload
- createBookInput
- createBookPayload

```
1 {
2   books(first: 3) {
3     edges {
4       node {
5         title
6         author
7         publicationDate
8       }
9     }
10  }
11 }
```

Variables Headers

```
{
  "data": {
    "books": {
      "edges": [
        {
          "node": {
            "title": "Quia Vel Omnis Reprehenderit",
            "author": "Walker Weber",
            "publicationDate": "1989-02-26"
          }
        },
        {
          "node": {
            "title": "Qui Blanditiis Sed Harum",
            "author": "Doris Funk I",
            "publicationDate": "2014-02-07"
          }
        },
        {
          "node": {
            "title": "Similique Minus Sit Soluta",
            "author": "Brenden Deckow",
            "publicationDate": "1981-07-07"
          }
        }
      ]
    }
  }
}
```

# Hiding Fields, The Laravel Way

```
namespace App\Models;

use ApiPlatform\Metadata\ApiResource;
use Illuminate\Database\Eloquent\Model;

#[ApiResource]
class Book extends Model {
    //The attributes that should be hidden (deny list).
    protected $hidden = ['isbn'];
}
```

# Or Using an Allow List

```
namespace App\Models;

use ApiPlatform\Metadata\ApiResource;
use Illuminate\Database\Eloquent\Model;

#[ApiResource]
class Book extends Model {
    // The attributes that should be visible (allow list).
    protected $visible = ['title', 'description'];
}
```

# Adding Filters

```
namespace App\Models;

use ApiPlatform\Laravel\Eloquent\FILTER\PartialSearchFilter;
use ApiPlatform\Metadata\{ApiResource, QueryParameter};
use Illuminate\Database\Eloquent\Model;

#[ApiResource]
#[QueryParameter(
    key: 'title',
    filter: PartialSearchFilter::class,
)]
class Book extends Model {}
```

API Platform - API Platform x +

127.0.0.1:8000/api/

API PLATFORM

title Book title

string (query)

Minima

Execute Clear

Responses

Curl

```
curl -X 'GET' \  
'http://127.0.0.1:8000/api/books?page=1&title=Minima' \  
-H 'accept: application/ld+json'
```

Request URL

```
http://127.0.0.1:8000/api/books?page=1&title=Minima
```

Server response

Code	Details
200	<p>Response body</p> <pre>{   "@context": "/api/contexts/Book",   "@id": "/api/books",   "@type": "Collection",   "totalItems": 3,   "member": [     {       "@id": "/api/books/12",       "@type": "Book",       "id": 12,       "isbn": "9789993720478",       "title": "Quad Minima Dolore Eos",       "description": "Hic magnam quidem nihil. Voluptates corporis voluptas ut illo. Et ab porro cupiditate maiores voluptatem ipsum nihil. Voluptatem et quia aut dicta est omnis.",       "author": "Leanna Huels",       "publicationDate": "2002-04-02",       "createdAt": "2024-09-16T19:28:17+00:00",       "updatedAt": "2024-09-16T19:28:17+00:00"     },     {       "@id": "/api/books/32",</pre>

Minima

Tout surligner  Respecter la casse  Respecter les accents et diacritiques  Mots entiers Occurrence 1 sur 8 Bas de la page atteint, poursuite au début

# Built-in Filters

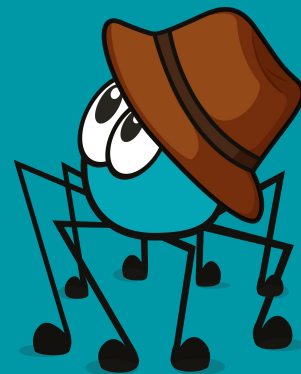
- ✓ **Text:** equals, partial matching, start, end
- ✓ **Date:** equals, greater, lower...
- ✓ **Boolean** logic: and, or
- ✓ **Ordering**
- ✓ **Range**
- ✓ Vulcain/GraphQL-like **“properties”** filter

Filter values are also validated!



# Configuring the API

- ✓ API Platform comes with a lot of settings
- ✓ Almost everything is configurable
- ✓ You can configure behavior:
  - Globally, in `config/api-platform.yaml`
  - For an endpoint, using `#[ApiResource]`
  - For an operation (e.g. `#[Post]`), using the operation attributes





# Config Example

```
#[ApiResponse(  
    paginationItemsPerPage: 10,  
    operations: [  
        // Read-only endpoint  
        new GetCollection(),  
        new Get(),  
    ],  
)]  
class Book extends Model {}
```

# Authentication

API Platform hooks into the native authentication mechanism, built-in support for:

- ✓ Standard Laravel Users
- ✓ Laravel Sanctum: SPA and simple API token
- ✓ Laravel Passport: OAuth server
- ✓ Laravel Socialite: OAuth Client (Facebook, X, LinkedIn, Google, GitHub, GitLab, Bitbucket, and Slack)



# Data Validation

## php artisan make:request

```
namespace App\Http\Requests;


class BookFormRequest extends
\Illuminate\Foundation\Http\FormRequest {
    public function authorize(): bool {
        return $this->user()->isAdmin();
    }

    public function rules(): array {
        return [
            'title' => 'required|unique:books|max:255',
            'description' => 'required',
            'author' => 'required|max:255',
        ];
    }
}
```

API Platform - API Platform

127.0.0.1:8000/api

API PLATFORM



Curl

```
curl -X 'POST' \  
  'http://127.0.0.1:8000/api/books' \  
  -H 'accept: application/ld+json' \  
  -H 'Content-Type: application/ld+json' \  
  -d '{  
    "isbn": "string",  
    "title": "",  
    "description": "",  
    "author": "",  
    "publicationDate": "2024-09-17T08:55:09.236Z",  
    "createdAt": "2024-09-17T08:55:09.236Z",  
    "updatedAt": "2024-09-17T08:55:09.236Z"  
  }'
```

Request URL

```
http://127.0.0.1:8000/api/books
```

Server response

Code Details

422 Error: Unprocessable Content

Response body

```
{  
  "@context": "/api/contexts/ValidationError",  
  "@id": "/api/validation_errors/57fc802eabc49055.jsonld",  
  "@type": "ValidationError",  
  "description": "The title field is required. (and 2 more errors)",  
  "type": "/validation_errors/57fc802eabc49055",  
  "title": "Validation Error",  
  "detail": "The title field is required. (and 2 more errors)",  
  "status": 422,  
  "violations": [  
    {  
      "propertyPath": "title",  
      "message": "The title field is required."  
    },  
    {  
      "propertyPath": "description",  
      "message": "The description field is required."  
    },  
    {  
      "propertyPath": "author",  
      "message": "The author field is required."  
    }  
  ]  
}
```

Download

# Authorization Logic

```
php artisan make:policy BookPolicy  
--model=Book
```

```
namespace App\Policies;  
  
use App\Models\Book;  
use App\Models\User;  
  
class BookPolicy {  
    public function viewAny(User $user): bool {}  
    public function view(User $user, Book $book): bool {}  
    public function create(User $user): bool {}  
    public function update(User $user, Book $book): bool {}  
    public function delete(User $user, Book $book): bool {}  
    public function restore(User $user, Book $book): bool {}  
    public function forceDelete(User $user, Book $book): bool {}  
}
```

# FrankenPHP

```
composer require laravel/octane  
php artisan octane:install --server=frankenphp  
php artisan octane:frankenphp
```



# Frontend

The Laravel variant is compatible with all API Platform JavaScript tools:

- ✓ Create Client: Next.js, Nuxt, Angular, Quasar and Vuetify PWA scaffolding
- ✓ Admin: React-Admin based admin UI



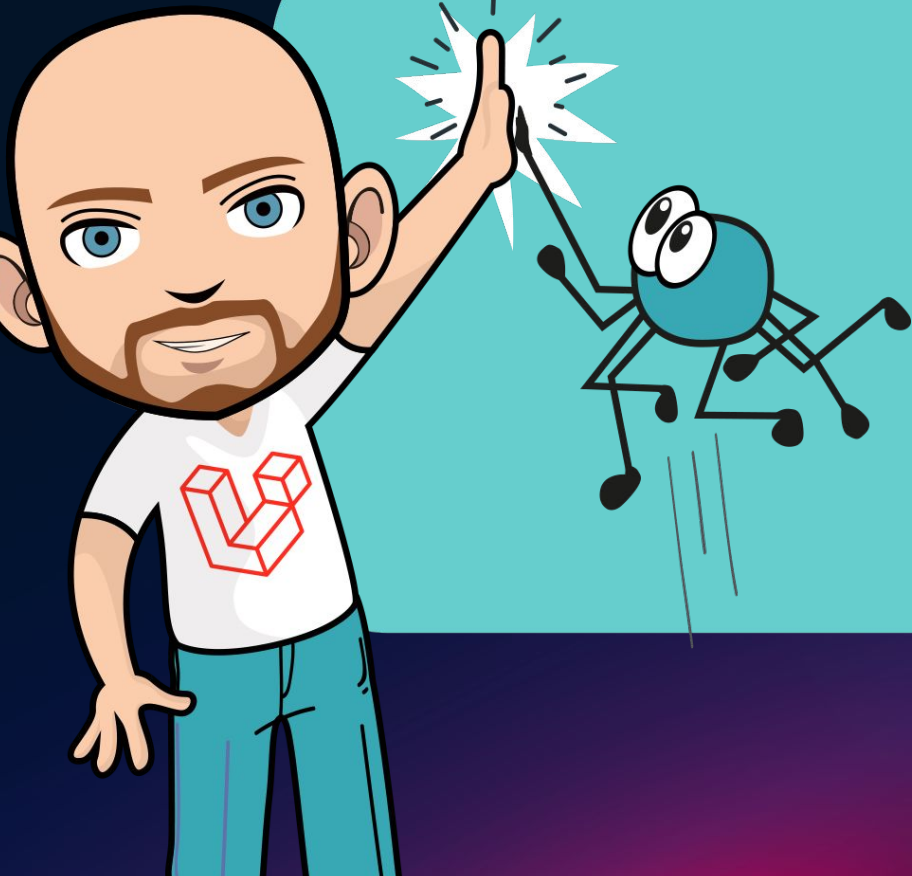
# Coming Soon: Real-time

---

- ✓ **Mercure** v1.0
- ✓ Native Laravel **Broadcast** integration
- ✓ Native Laravel **Echo** integration







”

We're so excited  
that API Platform  
now supports Laravel.

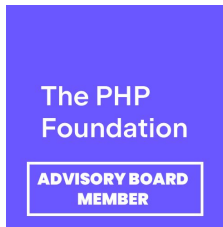
It's never been easier to create  
best-practice APIs using Laravel  
when paired with API Platform!

---

**TAYLOR OTWELL**  
AND THE LARAVEL TEAM



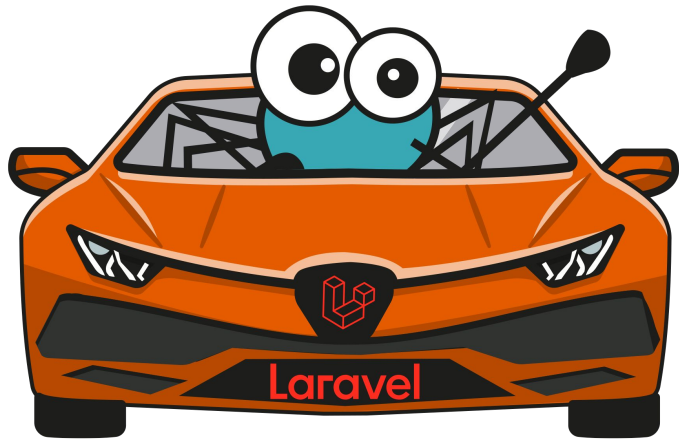
made with ❤️ by **Les-Tilleuls.coop**



Discover the new API Platform documentation  
[api-platform.com](https://api-platform.com)

# Thank you!

Any questions?



**FOLLOW ME!**

---

 @dunglas

 [dunglas.dev](https://dunglas.dev) / [les-tilleuls.coop](https://les-tilleuls.coop) / [api-platform.com](https://api-platform.com)

**API PLATFORM**  
CONFERENCE