
release v4.7.0 release v4.7.0 release v4.7.0 release v4.7.0 downloads 3.3M

Geocoder for Laravel

If you still use **Laravel 4**, please check out the `0.4.x` branch [here](#).

Version 4.0.0 is a backwards-compatibility-breaking update. Please review this documentation, especially the *Usage* section before installing.

This package allows you to use **Geocoder** in **Laravel 5**.

Requirements

- PHP >= 7.1.3
- Laravel >= 5.0

Installation

1. Install the package via composer:

```
1 composer require toin0u/geocoder-laravel
```

2. **If you are running Laravel 5.5 (the package will be auto-discovered), skip this step.** Find the `providers` array key in `config/app.php` and register the **Geocoder Service Provider**:

```
1 // 'providers' => [  
2     Geocoder\Laravel\Providers\GeocoderService::class,  
3 // ];
```

3. **Optional** I recommend adding the following lines to your `composer.json` file to prevent stale caches when upgrading or updating the package, both in your live and dev environments:

```
1     "post-update-cmd": [  
2         "@php artisan cache:clear",  
3     ],  
4     "post-install-cmd": [  
5         "@php artisan cache:clear",  
6     ]
```

Configuration

Pay special attention to the language and region values if you are using them. For example, the GoogleMaps provider uses TLDs for region values, and the following for language values: <https://developers.google.com/maps/faq#languagesupport>.

Further, a special note on the GoogleMaps provider: if you are using an API key, you must also use set HTTPS to true. (Best is to leave it true always, unless there is a special requirement not to.)

See the Geocoder documentation for a list of available adapters and providers.

Dedicated Cache Store *Recommended*

To implement the dedicated cache store, add another redis store entry in `config/database.php`, something like the following:

```
1  "redis" => [  
2      // ...  
3  
4      "geocode-cache" => [ // choose an appropriate name  
5          'host' => env('REDIS_HOST', '192.168.10.10'),  
6          'password' => env('REDIS_PASSWORD', null),  
7          'port' => env('REDIS_PORT', 6379),  
8          'database' => 1, // be sure this number differs from your  
                           other redis databases  
9      ],  
10 ]
```

You will also need to add an entry in `config/cache.php` to point to this redis database:

```
1  "stores" => [  
2      // ...  
3  
4      "geocode" => [  
5          'driver' => 'redis',  
6          'connection' => 'geocode-cache',  
7      ],  
8  ],
```

Finally, configure Geocoder for Laravel to use this store. Edit `config/geocoder.php`:

```
1  "cache" => [  
2      "store" => "geocode",  
3  
4      // ...  
5  ],
```

Disabling Caching on a Query-Basis You can disable caching on a query-by-query basis as needed, like so:

```
1 $results = app("geocoder")
2     ->doNotCache()
3     ->geocode('Los Angeles, CA')
4     ->get();
```

Providers

If you are upgrading and have previously published the geocoder config file, you need to add the `cache-duration` variable, otherwise cache will be disabled (it will default to a 0 cache duration). The default cache duration provided by the config file is 999999999 seconds, essentially forever.

By default, the configuration specifies a Chain provider, containing the GoogleMaps provider for addresses as well as reverse lookups with lat/long, and the GeoIP provider for IP addresses. The first to return a result will be returned, and subsequent providers will not be executed. The default config file is kept lean with only those two providers.

However, you are free to add or remove providers as needed, both inside the Chain provider, as well as along-side it. The following is the default configuration provided by the package:

```
1 use Geocoder\Provider\Chain\Chain;
2 use Geocoder\Provider\GeoPlugin\GeoPlugin;
3 use Geocoder\Provider\GoogleMaps\GoogleMaps;
4 use Http\Client\Curl\Client;
5
6 return [
7
8     /*
9     |-----
10    | Cache Duration
11    |-----
12    |
13    | Specify the cache duration in seconds. The default approximates a
14    | forever
15    | cache, but there are certain issues with Laravel's forever
16    | caching
17    | methods that prevent us from using them in this project.
18    | Default: 9999999 (integer)
19    | */
20    'cache-duration' => 9999999,
21
```

```

22  /*
23  |-----
24  | Providers
25  |-----

26  |
27  | Here you may specify any number of providers that should be used
28  | to
29  | perform geocaching operations. The `chain` provider is special,
30  | in that it can contain multiple providers that will be run in
31  | the sequence listed, should the previous provider fail. By
32  | default the first provider listed will be used, but you
33  | can explicitly call subsequently listed providers by
34  | alias: `app('geocoder')->using('google_maps')`.
35  | Please consult the official Geocoder documentation for more info.
36  | https://github.com/geocoder-php/Geocoder#providers
37  |
38  */
39  'providers' => [
40      Chain::class => [
41          GoogleMaps::class => [
42              env('GOOGLE_MAPS_LOCALE', 'us'),
43              env('GOOGLE_MAPS_API_KEY'),
44          ],
45          GeoPlugin::class => [],
46      ],
47  ],
48
49  /*
50  |-----

51  | Adapter
52  |-----

53  |
54  | You can specify which PSR-7-compliant HTTP adapter you would like
55  | to use.
56  | There are multiple options at your disposal: CURL, Guzzle, and
57  | others.
58  | Please consult the official Geocoder documentation for more info.
59  | https://github.com/geocoder-php/Geocoder#usage
60  | Default: Client::class (FQCN for CURL adapter)
61  |
62  */
63  'adapter' => Client::class,
64
65  /*

```

```
66 | -----
67 | Reader
68 | -----
69 |
70 | You can specify a reader for specific providers, like GeoIp2,
71 | which
72 | connect to a local file-database. The reader should be set to an
73 | instance of the required reader class or an array containing the
74 | reader
75 | class and arguments.
76 | Please consult the official Geocoder documentation for more info.
77 | https://github.com/geocoder-php/geoip2-provider
78 | Default: null
79 |
80 | */
81 | 'reader' => null,
82 |
83 | ];
```

Adapters

By default we provide a CURL adapter to get you running out of the box. However, if you have already installed Guzzle or any other PSR-7-compatible HTTP adapter, you are encouraged to replace the CURL adapter with it. Please see the Geocoder Documentation for specific implementation details.

Customization

If you would like to make changes to the default configuration, publish and edit the configuration file:

```
1 php artisan vendor:publish --provider="Geocoder\Laravel\Providers\
  GeocoderService" --tag="config"
```

Usage

The service provider initializes the `geocoder` service, accessible via the facade `Geocoder::...` or the application helper `app('geocoder')->...`

Geocoding Addresses

Get Collection of Addresses

```
1 app('geocoder')->geocode('Los Angeles, CA')->get();
```

Get IP Address Information

```
1 app('geocoder')->geocode('8.8.8.8')->get();
```

Reverse-Geocoding

```
1 app('geocoder')->reverse(43.882587,-103.454067)->get();
```

Dumping Results

```
1 app('geocoder')->geocode('Los Angeles, CA')->dump('kml');
```

Dependency Injection

```
1 use Geocoder\Laravel\ProviderAndDumperAggregator as Geocoder;
2
3 class GeocoderController extends Controller
4 {
5     public function getGeocode(Geocoder $geocoder)
6     {
7         $geocoder->geocode('Los Angeles, CA')->get()
8     }
9 }
```

Upgrading

Anytime you upgrade this package, please remember to clear your cache, to prevent incompatible cached responses when breaking changes are introduced (this should hopefully only be necessary in major versions):

```
1 php artisan cache:clear
```

1.x to 4.x

Update your composer.json file:

```
1 "toin0u/geocoder-laravel": "^4.0",
```

The one change to keep in mind here is that the results returned from **Geocoder for Laravel** are now using the Laravel-native Collections class instead of returning an instance of **AddressCollection**. This should provide greater versatility in manipulation of the results, and

be inline with expectations for working with Laravel. The existing `AddressCollection` methods should map straight over to Laravel's `Collection` methods. But be sure to double-check your results, if you have been using `count()`, `first()`, `isEmpty()`, `slice()`, `has()`, `get()`, or `all()` on your results.

Also, `getProviders()` now returns a Laravel Collection instead of an array.

Alert: if you have been using the `getIterator()` method, it is no longer needed. Simply iterate over your results as you would any other Laravel collection.

Deprecated: - the `all()` method on the geocoder is being deprecated in favor of using `get()`, which will return a Laravel Collection. You can then run `all()` on that. This method will be removed in version 5.0.0. - the `getProvider()` method on the geocoder is being deprecated in favor of using `getProviders()`, which will return a Laravel Collection. You can then run `first()` on that to get the same result. This method will be removed in version 5.0.0.

Added: this version introduces a new way to create more complex queries: - `geocodeQuery()` - `reverseQuery()`

Please see the Geocoder documentation for more details.

0.x to 1.x

If you are upgrading from a pre-1.x version of this package, please keep the following things in mind:

1. Update your `composer.json` file as follows:

```
1 "toin0u/geocoder-laravel": "^1.0",
```

2. Remove your `config/geocoder.php` configuration file. (If you need to customize it, follow the configuration instructions below.)
3. Remove any Geocoder alias in the aliases section of your `config/app.php`. (This package auto-registers the aliases.)
4. Update the service provider entry in your `config/app.php` to read:

```
1 Geocoder\Laravel\Providers\GeocoderService::class,
```

5. If you are using the facade in your code, you have two options:
 1. Replace the facades `Geocoder::` (and remove the corresponding `use` statements) with `app('geocoder')->`.
 2. Update the `use` statements to the following:

```
1 use Geocoder\Laravel\Facades\Geocoder;
```

6. Update your query statements to use `->get()` (to retrieve a collection of GeoCoder objects) or `->all()` (to retrieve an array of arrays), then iterate to process each result.

Troubleshooting

- Clear cache: `php artisan cache:clear`.
- If you are still experiencing difficulties, please open an issue on GitHub: <https://github.com/geocoder-php/GeocoderLaravel/issues>.

Changelog

<https://github.com/geocoder-php/GeocoderLaravel/blob/master/CHANGELOG.md>

Contributor Code of Conduct

Please note that this project is released with a Contributor Code of Conduct. By participating in this project you agree to abide by its terms.

License

GeocoderLaravel is released under the MIT License. See the bundled LICENSE file for details.