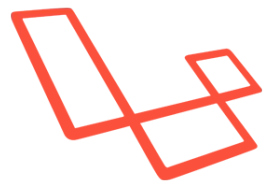


Building Advanced Instant Searches



Laravel

+



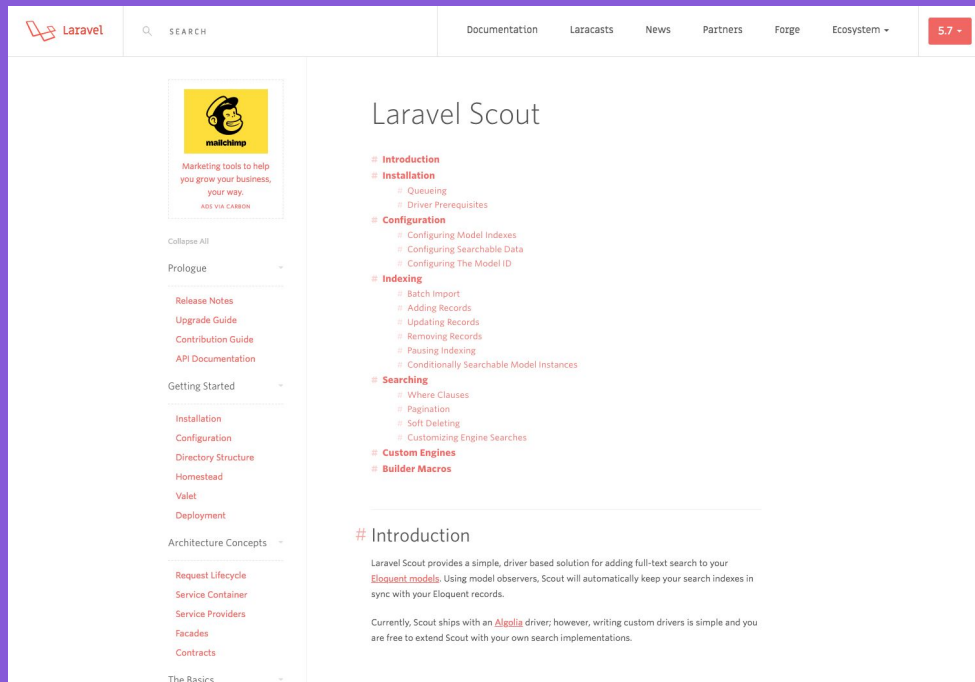
algolia



What is Scout?

Laravel Scout provides a full-text search that can be used with standard Eloquent models.

Currently built to support Algolia but custom drivers and direct database drivers are possible.



<https://laravel.com/docs/5.7/scout>

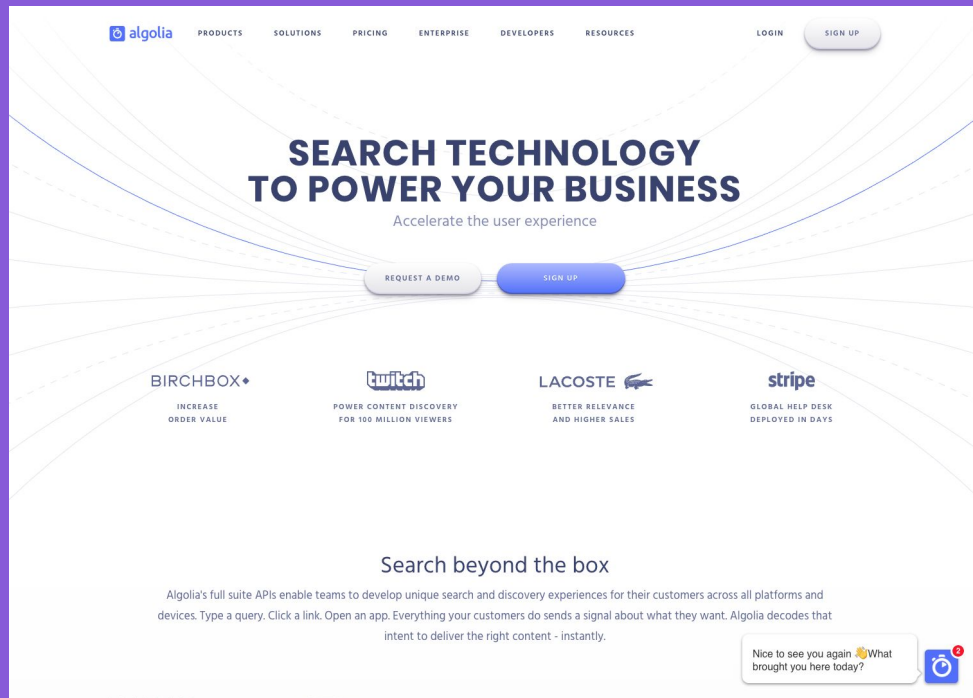
What is Algolia?

Algolia is a platform that stores key value indexes and provides instant searching with high relevance results

Algolia Includes:

- Direct API
- SDK for majority of languages
- Laravel package

<https://www.algolia.com>



Installing Scout & Algolia

Scout Requirements:

- PHP 7.1+
- Laravel 5.3+

Scout Extended Requirements:

- PHP 7.1+
- Laravel 5.6+

```
// Install Laravel Scout
composer require laravel/scout

// None extended
composer require algolia/algoliasearch-client-php:^1.27

// Scout Extended
composer require algolia/scout-extended

// Publish the provider to setup config
php artisan vendor:publish --provider="Laravel\Scout\ScoutServiceProvider"
```

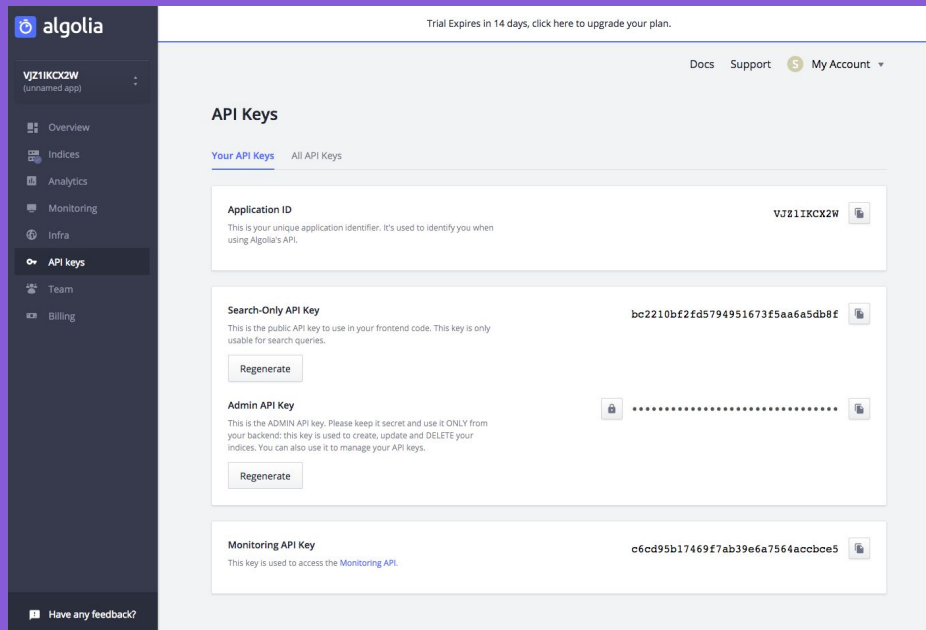


Configuration

You will need to register an account with Algolia to obtain your API keys

Once you have your keys simply drop them in as environment variables. They need to match what is inside
`/config/scout.php`

```
## Algolia Keys
ALGOLIA_APP_ID=#####
ALGOLIA_PUBLIC=#####
ALGOLIA_SECRET=#####
```



Indexing Models

Scout comes with simple traits you can just pull in on any model you want to be indexable.

The Searchable trait comes with observers that automatically push data back to Algolia upon changes

```
// Import all models
php artisan scout:import

// Single model import
php artisan scout:import "App\Location"
```

```
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;
use Laravel\Scout\Searchable;

class Location extends Model
{
    use Searchable;

    protected $fillable = [
        'name', 'address', 'description', 'latitude', 'longitude'
    ];
}
```

Customising Index Data

Scout allows you to alter the data before it is sent to the index with individual model override function `toSearchableArray()`.

You can use this to add extra attributes for the search index to help with keyword relevance.

You can also push in relational data that might be useful to the item.

```
/**
 * @return \Illuminate\Database\Eloquent\Relations\HasOne
 */
public function manager()
{
    return $this->hasOne('App\Manager', 'location_id');
}

/**
 * Get the indexable data array for the model.
 *
 * @return array
 */
public function toSearchableArray()
{
    $array = $this->toArray();

    // Applies Scout Extended default transformations:
    $array = $this->transform($array);

    // Add an extra attribute:
    $array['created_month'] = date('F', strtotime($array['created_at']));

    // Add relational data into index
    $array['manager_name'] = $this->manager->name;

    return $array;
}
```

Index Aggregators

An Aggregator allows you to combine multiple model indexes into one to create a global / cross index search.

Quicker than manually forcing models into a combined index.

Each model still exists as its own index.

```
// Create the Aggregator class
php artisan scout:make-aggregator News

// Configured Aggregator
class News extends Aggregator
{
    /**
     * The names of the models that should be aggregated.
     *
     * @var string[]
     */
    protected $models = [
        Blog::class,
        Event::class
    ];
}
```



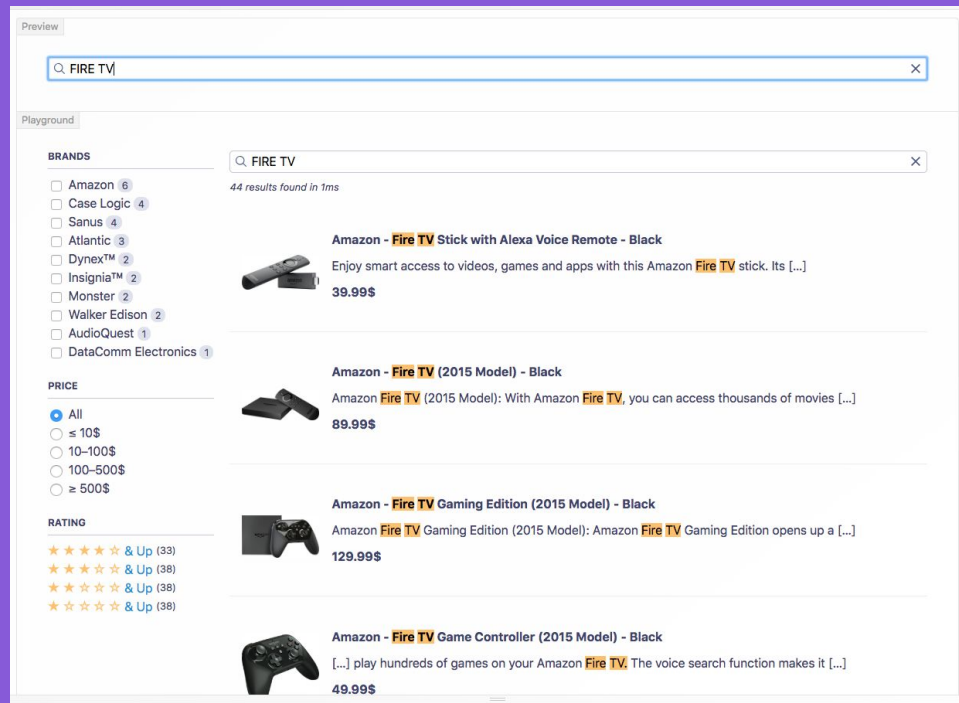
InstantSearch.js

Search of libraries that make it quick to get an instant / auto complete search going.

It is fully customisable so you have full control of your search and how the results are displayed

Also available in:

- React
- VueJS
- iOS
- Android



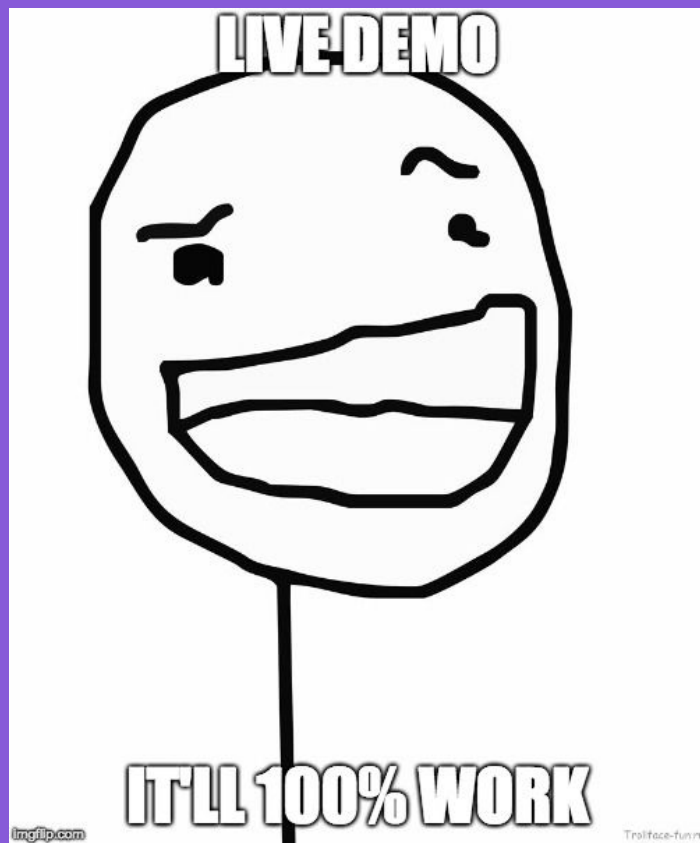
Backend Searching

Scout Extended adds its own set of Eloquent querying methods that you can attach. This includes Geolocation search.

The downside is that you can't attach normal text / string querying against specific attributes.

```
// Basic text query with Algolia backend
$results = Location::search('search query')->get();

// Geolocation querying backend
$results = Location::search($request->get('search'))
    ->aroundLatLng($request->get('latitude'), $request->get('longitude'))
    ->with( [
        'aroundRadius' => $request->get('radius')
    ])
    ->get();
```



Pricing

We have a voucher code that will give you 2 months free!

Voucher: **LARAVELEEDS**

Go to:
<https://www.algolia.com/redeem>

		
COMMUNITY	STARTER	PRO
Free for non-commercial use	Starting at \$29/mo	Starting at \$499/mo
50K operations 10K records	250K operations 50K records	5M operations 1M records
Not available	100k operations: \$10/month 20k records: \$10/month	100k operations: \$5/month 20k records: \$5/month

Quick Tips

1. Use `SCOUT_PREFIX` to create different indexes per environment
2. Use `php artisan scout:reimport` to avoid downtime
3. Be aware when the free trial ends your account stops working. You need to login and select community plan to keep it working or move to paid plan.



Thank you!

Github: <https://github.com/parallax/laravel-scout-algolia-demo>

Twitter: @jamiesefton / @parallax

Slides will be posted on meetup

PARALLAX

London
3rd Floor, 6 Duke Street
W1U 3EN

Leeds
The Old Brewery
LS2 7ES