**Tops Infosolutions**

Author: Prajwal Poojari

Elastic Search in Laravel

September 06, 2019

# By Two method

**1. Laravel scout**

Ref: <https://medium.com/@samogorm/basic-search-functionality-with-elasticsearch-laravel-scout-6ac182c99cbf>

-error in setup by this above url

You have installed [**ElasticSearch**](https://www.elastic.co/downloads/elasticsearch) to your machine

- from this doc on page 3

**Practically done & success in below url**

<https://code.tutsplus.com/tutorials/how-to-setup-a-full-text-search-using-scout-in-laravel--cms-30702>

**2. by prajwal document**

Ref:this doc

# Product Overview

Elasticsearch is a distributed, RESTful search and analytics engine that centrally stores your data so you can search, index, and analyze data of all shapes and sizes. Elasticsearch is a search engine based on the Lucene library. It provides a distributed, multitenant-capable full-text search engine with an HTTP web interface and schema-free JSON documents. Elasticsearch is developed in Java. Following an open-core business model, parts of the software are licensed under various open-source licenses (mostly the Apache License)[2], while other parts[3] fall under the proprietary (source-available) Elastic License. Official clients are available in Java, .NET (C#), PHP, Python, Apache Groovy, Ruby and many other languages.[4] According to the DB-Engines ranking, Elasticsearch is the most popular enterprise search engine followed by Apache Solr, also based on Lucene.[5]. This Product is developed by Elastic NV

MySQL > Databases > Tables > Columns/Rows

Elasticsearch > Indices > Types > Documents with properties

## 

## How to install into server (linux)

To install Elastic Search on Linux

wget -qO – https://artifacts.elastic.co/GPG-KEY-elasticsearch

sudo apt-key add –

sudo apt-get install apt-transport-https

echo "deb https://artifacts.elastic.co/packages/5.x/apt stable main" | sudo tee -a /etc/apt/sources.list.d/elastic-5.x.list

sudo apt-get update && sudo apt-get install elasticsearch

Access by http://localhost:9200

### 

### To make it accessible from localhost to any ip i.e 192.168.xx.xxx

sudo nano /etc/elasticsearch/elasticsearch.yml

edit this

network.host: 192.168.xx.xxx

sudo service elasticsearch restart

**To start or stop the server, following commands**

sudo service elasticsearch start

sudo service elasticsearch stop

## Laravel Package used to integrate elastic search

An easy way to use the official ElasticSearch client in your Laravel applications.

Installation guide and package available on github: <https://github.com/cviebrock/laravel-elasticsearch>

# Usage

Every Eloquent module contains its index and type which are two important required params of Elastic Search (considering a **Company** model)

class Company extends Eloquent

{

const ELASTIC\_INDEX = 'companies';

const ELASTIC\_TYPE = 'company';

protected $casts = [

'year' => 'int',

'rating' => 'float'

];

protected $fillable = [

'name',

'description',

'year',

'rating',

'image'

];

}

Now inserting value in elastic search. Example, whenever a company is inserted in your mysql table, we will insert into Elastic Search. See following example

In Laravel Controller

Using libraries

use Elasticsearch\ClientBuilder;

Constructing with host and port

public function \_\_construct()

{

$this->client = ClientBuilder::create()->setHosts([env('ELASTICSEARCH\_HOST').':'.env('ELASTICSEARCH\_PORT')])->build();

}

In .env

ELASTICSEARCH\_HOST=192.168.4.41

ELASTICSEARCH\_PORT=9200

ELASTICSEARCH\_SCHEME=http

ELASTICSEARCH\_USER=

ELASTICSEARCH\_PASS=

Now insert index method of elastic search

$data = [

'body' => [

'id' => $company->id,

'name' => $company->name,

'number' => $company->number,

'careof' => $company->careof,

],

'index' => Company::ELASTIC\_INDEX,

'type' => Company::ELASTIC\_TYPE,

'id' => $company->id

];

$response = $this->client->index($data);

/\* For All inserted data visible

http://192.168.4.152:9200/your\_index\_name/\_search

\*/

To check if already exists before insert

$data = [

'index' => Company::ELASTIC\_INDEX,

'type' => Company::ELASTIC\_TYPE,

'id' => $company->id

];

$checkdata = $this->client->exists($data);

**To update an entry**

$data = [

'body' => [

'doc' => [

'id' => $company->id,

'name' => $company->name,

'number' => $company->number

]

],

'index' => Company::ELASTIC\_INDEX,

'type' => Company::ELASTIC\_TYPE,

'id' => $company->id

];

$this->client->update($data);

**To delete an entry**

$data = [

'index' => Company::ELASTIC\_INDEX,

'type' => Company::ELASTIC\_TYPE,

'id' => $id

];

$response = $this->client->delete($data);

Finally the main search part. How search works

$params = [

'index' => Company::ELASTIC\_INDEX,

'type' => Company::ELASTIC\_TYPE,

'body' => [

'query' => [

'bool' => [

'should' => [

['match' => [

'name' => [

'query' => $text,

'fuzziness' => '1'

]

]],

['match' => [

'number' => [

'query' => $text,

'fuzziness' => '0'

]

]],

['match' => [

'country' => [

'query' => $text,

'fuzziness' => '0'

]

]]

]

],

],

"sort" => [

[$sortby => $sorttype],

'\_score'

],

"from" => $start\_from,

"size" => $size,

]

];

Above contains **from** and **size.** These are take and limit of Laravel queries. Now you can sort by the field name and direction. By default sorting is only enabled for numeric fields. To sort on a text fields, you have to call following request

<http://192.168.4.41:9200/companies/_mapping/company>

{

"company": {

"properties": {

"name": {

"type": "text",

"fielddata": true

}

}

}

}

**“name”** is the field name and companies and company in url are the index and type as mentioned before.

The query field is in detail in documentation of elastic search

<https://www.elastic.co/guide/en/elasticsearch/reference/current/query-filter-context.html>

To delete whole index,

DELETE <http://192.168.4.41:9200/companies/>

To increase maximum result window (default is 10000, **from** key value)

PUT http://192.168.4.41:9200/companies/\_settings

{

"index.max\_result\_window": 500000

}

Now Elastic Search has its own implementation of view and manipulating data named Kibana

Installing kibana

wget<https://artifacts.elastic.co/downloads/kibana/kibana-5.6.16-amd64.deb>

[sudo dpkg -i kibana-5.6.16-amd64.deb](https://artifacts.elastic.co/downloads/kibana/kibana-5.6.16-amd64.deb)

sudo nano /etc/kibana/kibana.yml

sudo systemctl daemon-reload

sudo systemctl enable kibana

sudo systemctl start kibana

Access by [http://localhost:5601](http://localhost:5601/)

[Before installing elastic search or kibana please go through the](http://localhost:5601/)

Compatibility version

[https://www.elastic.co/support/matrix#matrix\_compatibility](https://www.elastic.co/support/matrix" \l "matrix_compatibility)

References:

<https://factory.hr/blog/elasticsearch-introduction-implementation-and-example>

<https://medium.com/@babenko.i.a/how-to-make-laravel-and-elasticsearch-become-friends-55ed7690331c>

<https://madewithlove.be/how-to-integrate-your-laravel-app-with-elasticsearch/>

Compatibility Matrix

[https://www.elastic.co/support/matrix#matrix\_compatibility](https://www.elastic.co/support/matrix" \l "matrix_compatibility)