

|  |  |
| --- | --- |
| **Document title** | **Elasticsearch** |
| **Document reference number** |  |
| **Project name** | **UNISA e-Content** |
| **Unisa Department** | **Department of Economics** |
| **Developer** | **Dario Alfredo** |

|  |  |  |
| --- | --- | --- |
|  |  |  |

Table of Contents

[Description of Tool (Elasticsearch) 5](#_Toc491939331)

[This Elasticsearch system is a tool that uses CURL to post a JSON object to an Elasticsearch server. 5](#_Toc491939332)

[How to implement the Elasticsearch Tool 5](#_Toc491939333)

[Step 1.1 5](#_Toc491939334)

[Step 1.2 5](#_Toc491939335)

[Elasticsearch::search() parameter/arguments requirements 5](#_Toc491939336)

[The parameters 5](#_Toc491939337)

[The arguments 5](#_Toc491939338)

[Configuration: 5](#_Toc491939339)

[The .env file 5](#_Toc491939340)

[4 How to use the Elasticsearch Tool 6](#_Toc491939341)

[5 Technical Aspects 6](#_Toc491939342)

[6 Pre-requisites 6](#_Toc491939343)

[7 Running it 6](#_Toc491939344)

[8 Code Snippets 7](#_Toc491939345)

[8.1 Elasticsearch Server settings 7](#_Toc491939346)

[8.2 Elasticsearch Server mappings 7](#_Toc491939347)

[8.3 Elasticsearch query for courses (title, description) 8](#_Toc491939348)

[8.4 Elasticsearch class Implementation 8](#_Toc491939349)

Document sign-off

This document has been seen and accepted by the following people:

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Title | Date | Signature |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Revision history

|  |  |  |  |
| --- | --- | --- | --- |
| Revision | Compiler | Purpose | Change Summary |
| <yyyymmdd> | Name Surname | Document Created,  Internal Review,  External Review,  etc | Initial release,  Updates based on internal review,  Updates to section xxx,  etc |
|  |  |  |  |
|  |  |  |  |

# Description of Tool (Elasticsearch)

# This Elasticsearch system is a tool that uses CURL to post a JSON object to an Elasticsearch server.

# How to implement the Elasticsearch Tool

### Step 1.1

In your application, inside the script where you need to use this functionality, simply import the ‘Elasticsearch’ class. Take note of the namespace - which is App\Tools\Elasticsearch.

The import assumes that the class has been autoloaded by Composer in composer.json.

### Step 1.2

In your function/method call simply Inject the ‘Elasticsearch’ class and call its **search()** method.

# Elasticsearch::search() parameter/arguments requirements

### The parameters

The search() method definition has 3 parameters, namely: $query, $from and $size. The $from and $size have default values of 0 and 10 respectively.

### The arguments

The **$query** (json string): The $query argument must be a well-formed JSON string. The CURL handler posts this JSON to the Elasticsearch Server. Additionally this JSON query will differ in different contexts.

The **$from** (integer): The $from argument in conjunction with the $size argument is what enables the Elasticsearch server to return chunks of documents/rows per page. As the user cycles thru, the from parameter defines the offset from the first result to fetch, defaulting to 0.

The **$size** (integer): The size simply defines the number of documents/rows per page. The size argument defaults to 10.

## Configuration:

### The .env file

Define the Elasticsearch server end-point/URL in the .env file and use it in the ‘Elasticsearch’ class, as the URL to which the CURL call will be made to.

4 How to use the Elasticsearch Tool

The Elasticsearch system is very fast and flexible from the user perspective. The user gets results as he types or by clicking a search button. To get instant results simply enter the search term in an input box which will pop down the results if any.

Additionally, after entering the search term followed by a click of a search button, the app redirects to a page that contains the results. The results can be cycled thru back and forth.

5 Technical Aspects

**In the front-end, the Elasticsearch system works with a JavaScript library called Typeahead and Bloodhound. The system makes use of AJAX requests as well as normal requests. When request is AJAX, Typeahead and Bloodhound work to provide search-as-you-type results. Otherwise, a search button may be clicked which redirects to search output page.**

6 Pre-requisites

**Do a pull to update your local repo with the files/scripts required to run Elasticsearch. To avoid ‘Class not found’ issues, ensure that the class is autoloaded and properly name spaced when you use it.**

**In the command line terminal, enter the following Composer command in the root of the project:**

composer dump-autoload -o

7 Running it

To test this particular implementation, login as a student like Sue and visit <http://localhost:8000/lti/courses>. This page will show the input box and search button.

You might have to replace the ‘localhost:8000’ with your local environment.

Enjoy!!!

8 Code Snippets

### 8.1 Elasticsearch Server settings

{

"settings": {

"analysis": {

"filter": {

"autocomplete\_filter": {

"type": "edge\_ngram",

"min\_gram": 3,

"max\_gram": 20

}

},

"analyzer": {

"autocomplete": {

"type": "custom",

"tokenizer": "standard",

"filter": [

"lowercase",

"autocomplete\_filter"

]

}

}

}

}

}

### 8.2 Elasticsearch Server mappings

{

"mappings": {

"courses": {

"dynamic": "true",

"properties": {

"suggest": {

"type": "completion",

},

"id": {

"type": "int"

},

"title": {

"type": "text",

"analyzer": "autocomplete"

},

"description": {

"type": "text",

"analyzer": "autocomplete",

}

}

}

}

}

### 8.3 Elasticsearch query for courses (title, description)

{"query":{"function\_score":{"query":{"bool":{"must":[{"multi\_match":{"fields":["title^10","description^5"],"type":"cross\_fields","query":"english","minimum\_should\_match":"2<-1 5<70%"}}]}}}}}

### 8.4 Elasticsearch class Implementation

<?php

/\*\*

\* User: Dario.Alfredo

\* Date: 8/17/2017

\* Time: 9:57 AM

\*/

namespace App\Tools\Elasticsearch;

class Elasticsearch

{

public function search($query, $from = 0, $size = 10)

{

$from = $from === null ? 0 : $from;

$size = $size === null ? 10 : $size;

$search = 'http://localhost:9200/u-index/courses/\_search?from=' . $from . '&size=' . $size;

$ch = curl\_init();

curl\_setopt($ch, CURLOPT\_URL, $search);

curl\_setopt($ch, CURLOPT\_RETURNTRANSFER, true);

curl\_setopt($ch, CURLOPT\_CUSTOMREQUEST, "POST");

curl\_setopt($ch, CURLOPT\_POSTFIELDS, $query);

curl\_setopt($ch, CURLOPT\_RETURNTRANSFER, true);

curl\_setopt($ch, CURLOPT\_HTTPHEADER,array(

'Content-Type: application/json',

'Content-Length: ' . strlen($query)

)

);

$result = curl\_exec($ch);

curl\_close($ch);

return $result;

}

}