

# UNIVERSAL SEARCH

PREVIEW RELEASE



# Welcome

The basic search capability in Rock is quite powerful, but sometimes you may need more. That's where Universal Search comes in. Universal Search allows you to search multiple types of data at once in a full-text manner. In a sense, it's like Google for Rock. With this great power comes some additional technical knowledge, but don't worry we'll unpack it all here in this guide.

## Preview State

As of Rock v7 the Universal Search is in what we call a *Preview State*. That means that while it is feature-complete, things may change and adjust. Also, we're seeking your input on tweaks to the system to ensure everything is how it should be.

# Overview

The first thing that you'll need to enable Universal Search is a search provider. All data in Rock is stored in a SQL Server relational database. Relational databases are great for storing and retrieving large amounts of data stored as records, but they're not so great for doing full-text search. Rock's Universal Search feature relies on a different type of software for hosting the search. While we hope to have multiple options in the future, we currently only support one - Elasticsearch.

## Elasticsearch Provider

Elasticsearch is an open-source search index used by many of the largest websites including Netflix, EBay and the New York Times. It's free to use but does require that you have the ability to install software on your server. We provide a guide to installing Elasticsearch in the appendix of this guide. You can also choose to use Amazon's Elasticsearch service (this has not been fully regression-tested, so you might run into issues).

## Understanding How Universal Search Works

Rock is made up of numerous types of data called entities (at the time of this writing there are 376 types of entities). Entities are things like Person, Group, Group Member, etc. Universal Search supports indexing the following entity types (more could be added in the future based on community response).

- Person
- Business
- Group
- Content Channel Items
- Site

You might ask, why not index all entity attributes? Working with search is a delicate balance between providing the information the user wants and overloading them with random results. For instance, does anyone want to see results for Workflow Action Types in their search response?

### Custom Entities Are Supported

As a developer, you have the ability to enable your own entities to participate in Universal Search. More information on this will be coming in the future.

# Getting Started

Ready... set... let's get started! Assuming you have Elasticsearch installed, you are ready to get started.

## Enable a Search Provider

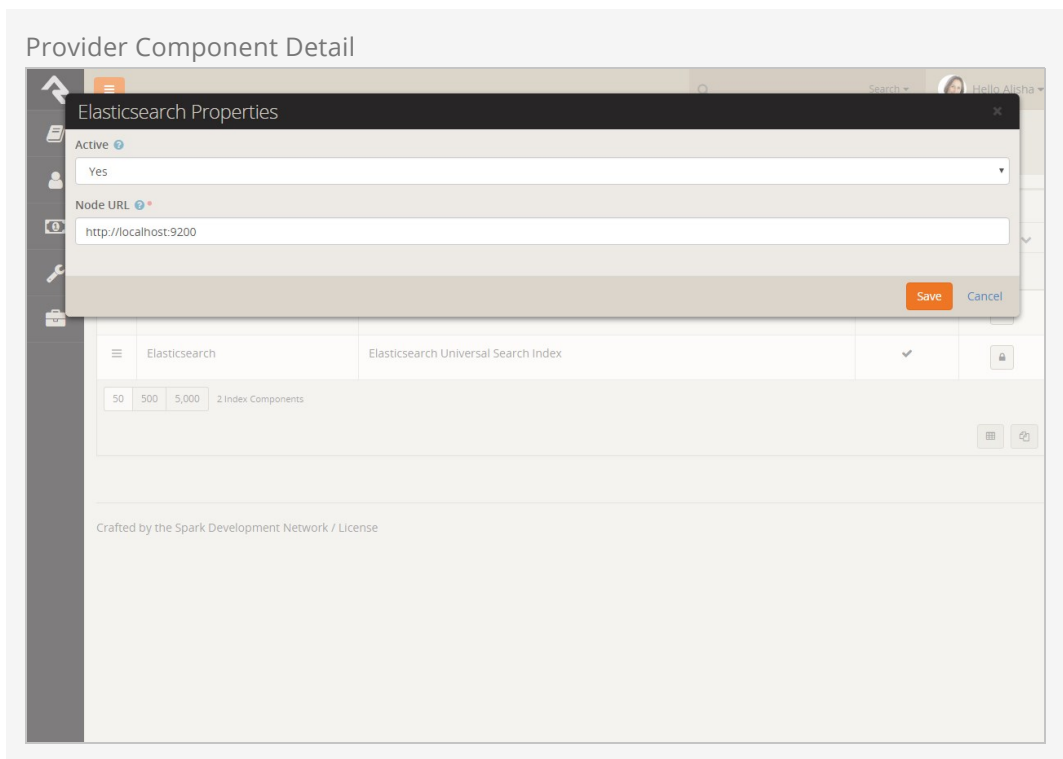
First, we'll need to tell Rock which search provider we'd like to use and provide the configuration details needed to connect. We do this under [Admin Tools > System Settings > Universal Search Index Components](#).

The screenshot displays the 'Provider Component List' page. At the top, there is a navigation bar with a search icon and a user profile labeled 'Hello Alisha'. Below the navigation bar, the page title is 'Universal Search Index Components' with a breadcrumb trail: 'Home / System Settings / Universal Search Index Components'. The main content area is titled 'Component List' and features a table with the following data:

Name	Description	Active	
Elasticsearch AWS	Amazon Elasticsearch Universal Search Index		<input type="checkbox"/>
Elasticsearch	Elasticsearch Universal Search Index	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Below the table, there are pagination controls showing '50', '500', and '5,000' items per page, and a summary of '2 Index Components'. At the bottom of the page, it says 'Crafted by the Spark Development Network / License'.

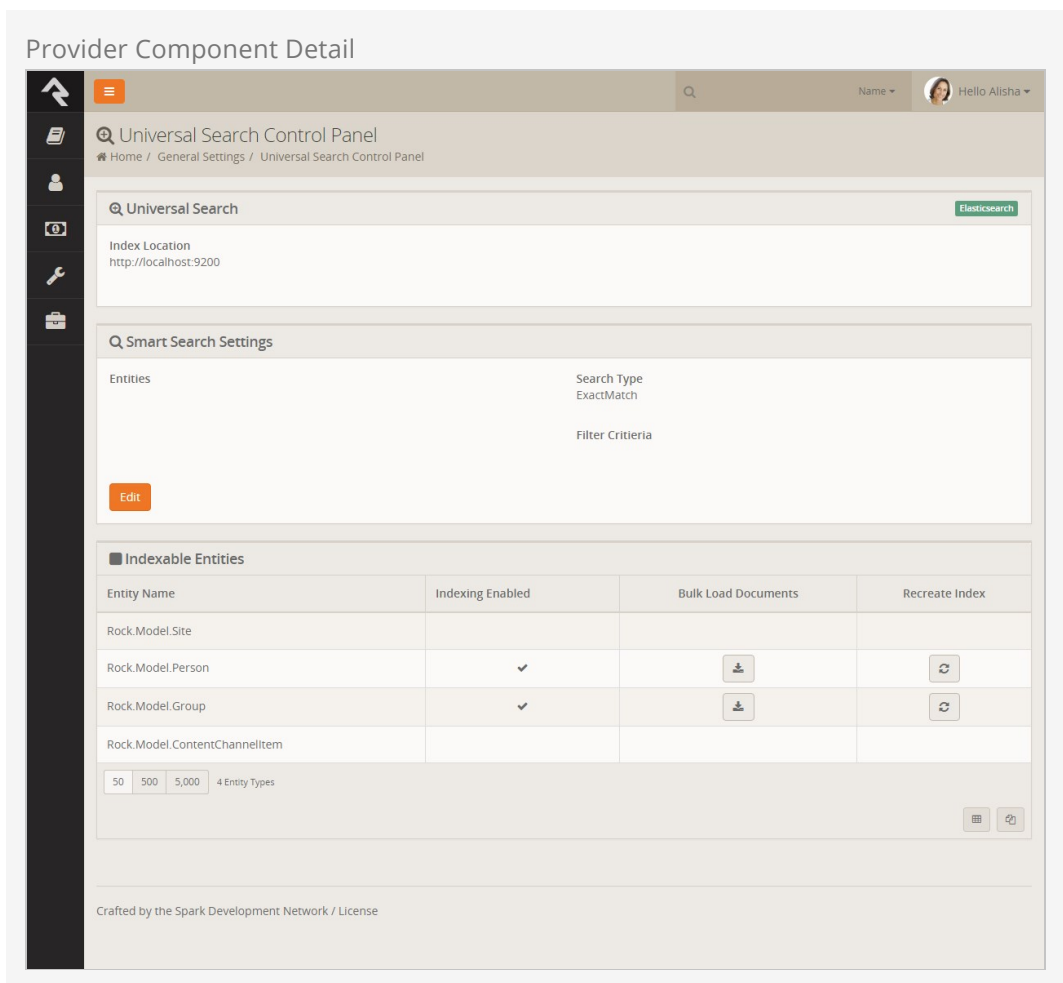
Next, select the provider type you'd like to enable.



## Enabling Entities for Universal Search



Now that we have a provider configured, we're ready to enable entities to be indexed.

To do this, navigate to `Admin Tools > General Settings > Universal Search Control Panel`.



At the top of this page you'll see a few details about the provider you selected. Below that you'll find a list of the entities that are able to be indexed.

To enable a new entity type, click the row of the entity and select `Enable Indexing` on the dialog that pops up. This entity is now enabled for Universal Search and a request has been queued to bulk index this entity.

	This button is will do a bulk re-index of all the documents.
	This button will delete the index and re-create it. This is good if you want to ensure all of the previous attributes have been purged from the index.

## Keeping Entities in Sync

As you'll read later, the entities should keep the index up-to-date as changes are made. A system job has also been created to re-index every night (this ensures that items added through SQL or other means are addressed). The settings of this job can be adjusted under `Admin Tools > System Settings > Jobs > Universal Search Re-Index`.

### Site Entity

This job will keep all entities except the Site indexed. The Site entity is a bit different and is discussed in detail below.

# Specifics for Entities

## Person

The person entity is pretty basic. Once enabled, all individuals in the database will be sent to the index. You can add specific person attributes to be indexed as well ( [Admin Tools > General Settings > Person Attributes](#) ). When you add/delete attributes to the index, you'll want to run a bulk load on the Person index to ensure they are available right away. Otherwise you'll need to wait until the bulk re-index happens that night. Don't forget; bulk loading is done from [Admin Tools > General Settings > Universal Search Control Panel](#) .

### Security Warning

Be careful when indexing person attributes because attribute security is not available in Universal Search.

## Business

Businesses don't have their own entity. In Rock business actually share the Person entity (which makes sense when you consider how businesses relate to things like giving, etc.) In Universal Search we do break them out so they look like a separate entity to allow you better filter options (sometimes you want businesses in your results; sometimes you don't) but when it comes to setting up indexing and settings, remember that they share with people.

## Group

When you enable groups to be indexed, you may wonder why nothing is displayed. Before groups are displayed, you'll need to configure which group types you want added to the index. You can configure this with the Group Type Editor ( [Admin Tools > General Settings > Group Types](#) ). Once you enable a group type, a call will be sent to immediately start bulk updating the groups (no extra steps required on your part).

When new groups are added or an existing group is modified, it will be immediately sent to the index.

You can also determine which group attributes you would like indexed. Changing the attributes will require a manual index re-load.

Applying a custom icon to the group type is a good idea, as it will be displayed in the



results.

## Content Channel Items

Content Channel Items work much the same way as groups. You'll need to configure which content channels you'd like to be indexed. You can also determine which attributes to include in the index. One other property to note is the *Content Channel Item Publishing Point* for the channel. This setting allows you to determine how to link to the resulting item. When left blank, the internal channel item page will be used. More commonly though, you'll want provide a template to a custom page (like a blog detail page).

## Site

Sites are quite different from the other entities above. When you index a site, you are really interested in searching all of the pages of the site. To do this, one must crawl the site looking for all of the content of the pages. For instance, a single *Blog Item Detail* page will need to be crawled several times to retrieve each of the blog posts. When you enable the Site entity, you'll need to complete the following steps:

1. Enable each site that you would like to add to the index under [Admin Tools > CMS Configuration > Sites](#).
2. Once enabled, be sure to add the *Index Starting Location* for the site. This is the URL that the crawler will use to start the index process.
3. Set up a new Rock Job to crawl the site on a schedule that you determine is best for your needs. This can be done under [Admin Tools > System Settings > Jobs](#). The job type will be *Rock.Jobs.IndexRockSite*.

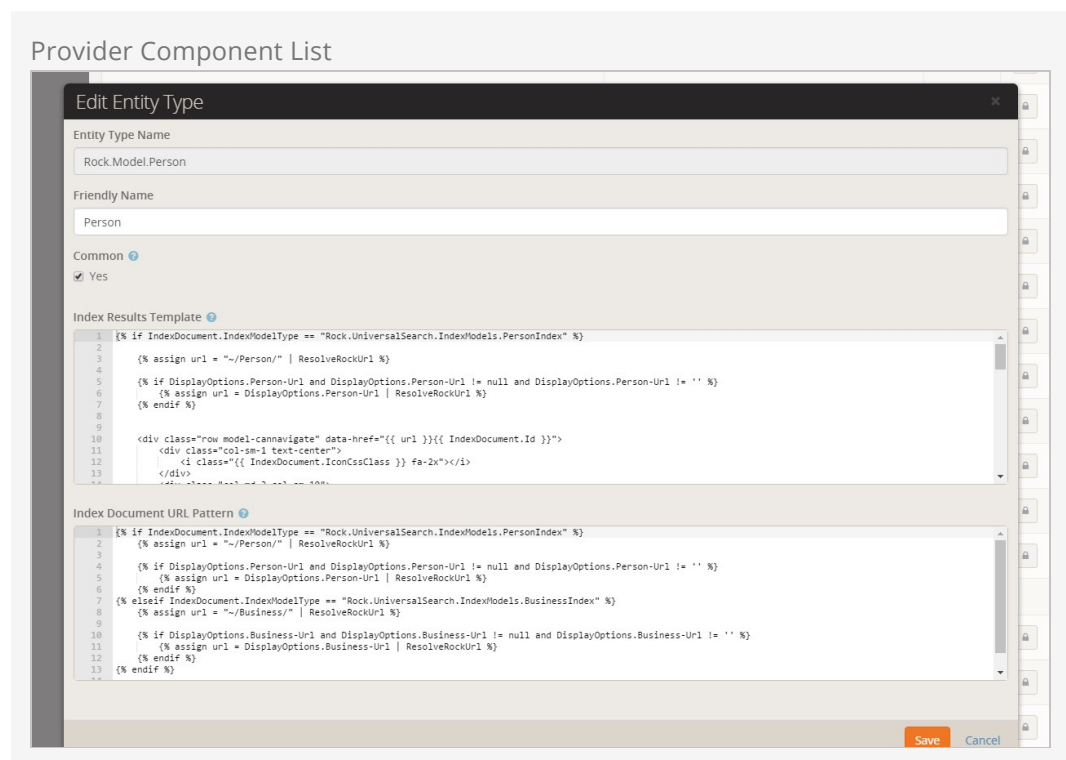
# Customizing Results for Entities

How results are returned from the search is important. Luckily, there are numerous ways to customize the results from the search. We cover all the options below.

## Default Entity Results

Each entity has a default result template that you can change. This is a great place to modify what you'd like to be returned across multiple search interfaces. You can edit these templates on a per-entity basis under [Admin Tools > Security > Entity Administration](#).

There are two templates available for your customization. The first, *Index Results Template*, is the template for the result row of the search. The other, *Index Document URL Pattern*, is the Lava template for determining the URL for the specific result when used in the Smart Search.



Let's consider a use case for how you might want to modify these templates. Say you have a custom group type in Rock that you would like to display differently than the normal group result. Perhaps you'd like to add a specific attribute value for the group

and link to a different group detail page. This is all possible by updating these two Lava templates.

### Sample Results Template

```
{% assign url = "~/Group/" | ResolveRockUrl %}

{% assign groupId = IndexDocument.GroupTypeId %}
{% if groupId == 24 %}
  {% assign url = "~/Organization/" | ResolveRockUrl %}
{% endif %}

{% if DisplayOptions.Group-Url and DisplayOptions.Group-Url != null and DisplayOptions.Group-Url != '' %}
  {% assign url = DisplayOptions.Group-Url | ResolveRockUrl %}
{% endif %}

<div class="row model-cannavigate" data-href="{{ url }}/{{ IndexDocument.Id }}">
  <div class="col-sm-1 text-center">
    <i class="{{ IndexDocument.IconCssClass }}" fa-2x"></i>
  </div>
  {% if groupId == 24 %}
    <div class="col-sm-4">
      <strong>{{ IndexDocument.Name }}</strong> <small>{{ IndexDocument.GroupTypeName }}</small>
      {% if IndexDocument.Description != null and IndexDocument.Description != '' %}
        <br>
        {{ IndexDocument.Description }}
      {% endif %}
    </div>
    <div class="col-sm-7">
      {{ IndexDocument.MemberList }}
    </div>
  {% else %}
    <div class="col-sm-11">
      <strong>{{ IndexDocument.Name }}</strong> <small>{{ IndexDocument.GroupTypeName }}</small>
      {% if IndexDocument.Description != null and IndexDocument.Description != '' %}
        <br>
        {{ IndexDocument.Description }}
      {% endif %}
    </div>
  {% endif %}
</div>
```

Notice the checking of the group type to customize the URL and the results.

### Sample URL Template

```
{% assign url = "~/Group/" | ResolveRockUrl %}

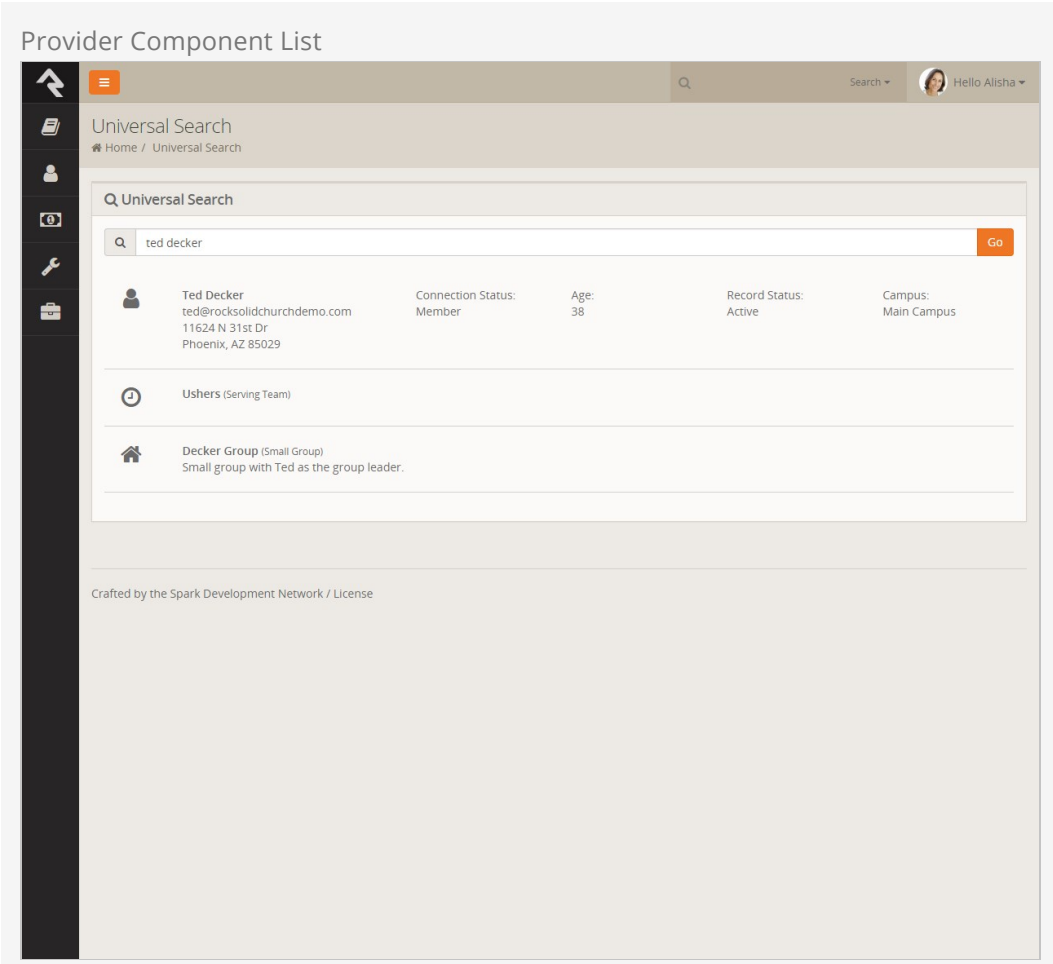
{% assign groupId = IndexDocument.GroupTypeId %}
{% if groupId == 24 %}
  {% assign url = "~/Organization/" | ResolveRockUrl %}
{% endif %}

{% if DisplayOptions.Group-Url and DisplayOptions.Group-Url != null and DisplayOptions.Group-Url != '' %}
  {% assign url = DisplayOptions.Group-Url | ResolveRockUrl %}
{% endif %}

{{ url }}/{{ IndexDocument.Id }}
```

## Custom Lava

When using the default search block, you can process your results through custom Lava you produce. This Lava is provided as a block setting. You'll need to both enable the usage of the custom Lava and provide a Lava template. A very basic template is provided as a starting point.



The screenshot displays a web interface titled "Provider Component List". The main content area is titled "Universal Search" and shows search results for the query "ted decker". The results are presented in a list format with icons and text. The first result is for "Ted Decker", showing contact information, connection status, age, record status, and campus. The second result is for "Ushers (Serving Team)" and the third is for "Decker Group (Small Group)".

Icon	Name	Connection Status	Age	Record Status	Campus
	Ted Decker ted@rocksolidchurchdemo.com 11624 N 31st Dr Phoenix, AZ 85029	Member	38	Active	Main Campus
	Ushers (Serving Team)				
	Decker Group (Small Group) Small group with Ted as the group leader.				

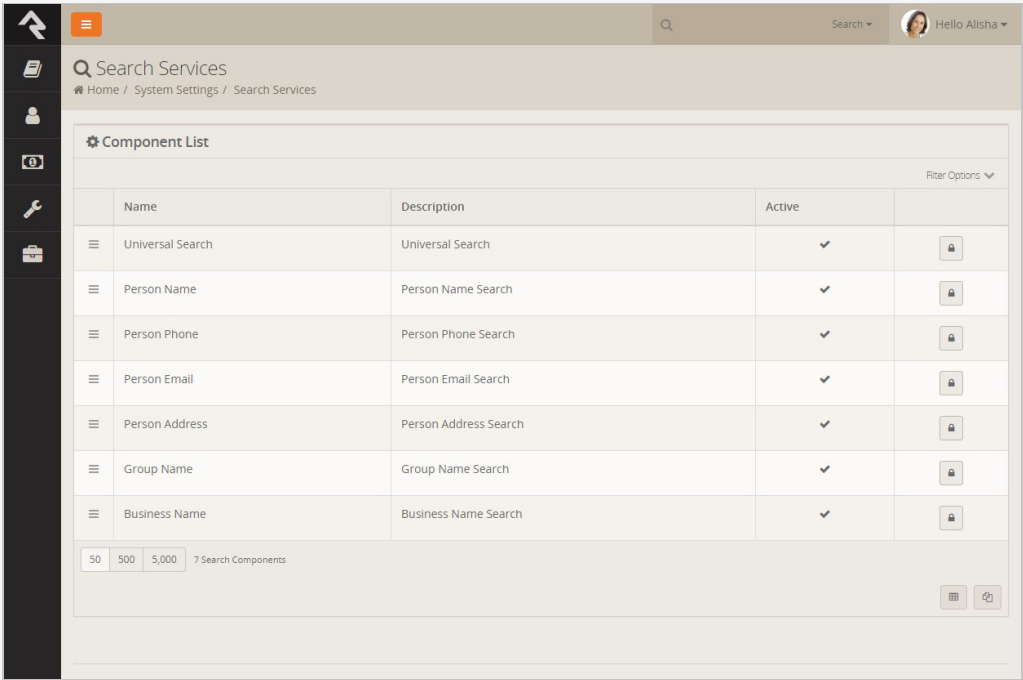
## Search Lava Command

When you want 100% control, you can drop down and use the search Lava command. This command works very similarly to the entity commands and allows you 100% control of the formatting of your results. See the documentation for this command in the Lava docs <http://rockrms.com/lava>.

# Smart Search Integration

If you've been using Rock for more than a day, you've used the Smart Search block at the top of the page. Universal Search can be configured to participate in Smart Search, and once it is, you'll find that it's your go-to search type.

Once you have Universal Search up and indexing, you'll need to enable the Smart Search integration. You'll do this under [Admin Tools > System Settings > Search Services](#).



The screenshot displays the 'Provider Component List' in the Rock CMS admin interface. The page title is 'Provider Component List' and the breadcrumb trail is 'Home / System Settings / Search Services'. The user is logged in as 'Hello Alisha'. The main content area shows a table of search components with the following data:

Name	Description	Active	
Universal Search	Universal Search	✓	🔒
Person Name	Person Name Search	✓	🔒
Person Phone	Person Phone Search	✓	🔒
Person Email	Person Email Search	✓	🔒
Person Address	Person Address Search	✓	🔒
Group Name	Group Name Search	✓	🔒
Business Name	Business Name Search	✓	🔒

At the bottom of the table, there are pagination controls showing '50', '500', and '5,000' items per page, and a total of '7 Search Components'. There are also icons for 'Add' and 'Refresh' at the bottom right of the table area.

Select the *Universal Search* item to configure its settings.

From here you'll provide the results URL. This should point to a page hosting the *Universal Search* block. The URL should be configured to append the *SmartSearch=true* parameter to tell the block the request came from the Smart Search feature. An example URL would be *universalsearch/{0}?SmartSearch=true*.

Additional Smart Search settings can be found on the Universal Search Control Panel ([Admin Tools > General Settings > Universal Search Control Panel](#)).

## Provider Component List

Home / General Settings / Universal Search Control Panel
Search ▼
Hello Alisha ▼

### Universal Search

Index Location  
http://localhost:9200

### Smart Search Settings

Search Entities ?

- Content Channel Item
- Group
- Person
- Site
- Merge Template Combined Person

Save Cancel

Search Type

Exact Match ▼

Field Criteria ?

### Indexable Entities

Entity Name	Indexing Enabled	Delete All Documents	Bulk Load Documents	Recreate Index
Rock.Model.Site	✓			
Rock.Model.Person	✓			
Rock.Model.Group				

Here you can enable which entities for which you'd like to show results, and provide advanced options.

# Installing Elasticsearch

To install Elasticsearch you will need to follow the steps below:

1. If your server does not have Java 8 or better, you will first need to install the Java Runtime Environment on your server. You can download the run-time from this link: <http://www.oracle.com/technetwork/java/javase/downloads/index.html>
2. Download Elasticsearch for Windows from <https://www.elastic.co/start>. At this point we are recommending you install version 2.x instead of v5.x.
3. Copy the files from the downloaded zip file to C:\Program Files\Elasticsearch-version
4. Configure Elasticsearch to run as a service using one of the options below:
  1. Version 2.x:  
<https://www.elastic.co/guide/en/elasticsearch/reference/2.3/setup-service-win.html>
  2. Version 5.x:  
<https://www.elastic.co/guide/en/elasticsearch/reference/current/windows.html>
5. Finally make sure that the newly configured service is started and has the *Startup Type* set to *Automatic*.

At this point you should have a running version of Elasticsearch. You can confirm this by entering: `http://localhost:9200`.

You should then see something like the text below in your web browser.

```
{
  "name" : "Tarantula",
  "cluster_name" : "elasticsearch",
  "version" : {
    "number" : "2.3.1",
    "build_hash" : "bd980929010aef404e7cb0843e61d0665269fc39",
    "build_timestamp" : "2016-04-04T12:25:05Z",
    "build_snapshot" : false,
    "lucene_version" : "5.5.0"
  },
  "tagline" : "You Know, for Search"
}
Get the message? Bingo!
```

### Security Note

Elasticsearch has no concept of security (yep you read that right). It is imperative that you place your Elasticsearch server behind a firewall and block access to port 9200 from the Internet.

## Elasticsearch HQ

If you're running version 2.x you may want to consider installing Elasticsearch HQ. It's a nice administrative console for Elasticsearch which gives you some visibility into what's going on under the hood. To do this simply:

1. Navigate to your elasticsearch/bin directory and type: `./plugin install royrusso/elasticsearch-HQ`
2. Open up your browser to: `http://localhost:9200/_plugin/hq/`