Adding Search To Your RockRMS Public Website With OpenSearchServer

A Powerful Open Source Solution

Let's cut to the chase. You want your public facing website to have search capabilities. This paper will detail how to use the great tool, OpenSearchServer, to provide that to your users. OpenSearchServer (OSS) is a very robust and feature-rich search tool. It can be used to crawl and index websites, databases, file systems, and more. We will be focusing on using OSS to provide search for your public-facing website published with RockRMS. This paper will give you the essential steps to set up and run OSS, as well as how to use the provided Rock Block to provide search in a seamless and integrated way.

Adding Search To Your RockRMS Public Website With OpenSearchServer

A Powerful Open Source Solution

Installation

As an alternative to installing OSS yourself, OpenSearchServer offers paid hosting solutions. If you choose this route, skip to the **Configuration** section.

OSS requires Java 1.7 or newer to be installed on the machine it is running on. Install, or upgrade, Java as needed. You can verify your Java version by running *'java –version'* at a command prompt.

- Download the OpenSearchServer .zip file from https://cloud.opensearchserver.com/opensearchserver#download and save it locally.
- Unzip the contents of this file into a folder on the computer you will be running OSS on.
    - You can run OSS on the same server running Rock. OSS uses port 9090 by default, so it can run side by side with IIS running Rock on 80/443.
    - Depending on the size of your website, the frequency of your crawls/indexing, etc. you may want to run OSS on its own computer.
- Run the start.bat file. This will open a command prompt and start the app.

Running OSS as a Service

In production, you will likely not want to have to log in to your OSS server and run a batch file.

Happily, you can configure things so that OSS runs as a service on your machine. The steps are simple.

- First, download the Apache Tomcat 8 Windows Service Installer
  http://tomcat.apache.org/download-80.cgi
- Install Apache Tomcat. On the configuration page, take careful note of the HTTP/1.1 Connector port. **This is now the port that OpenSearchServer is on.** In the example below I've set the port as 9090, to match what it would have been if I were running OSS from the batch file.
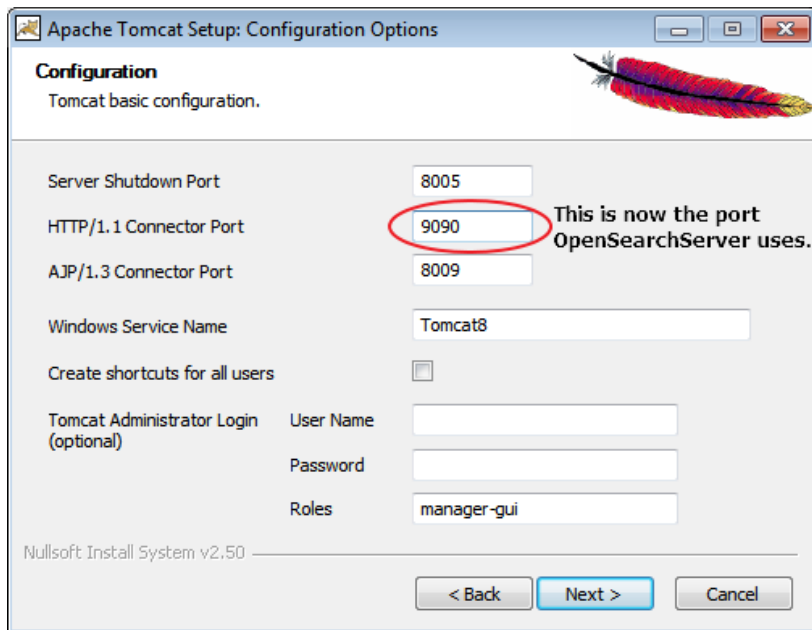
- Open your OpenSearchServer folder. From there, open the 'server' folder. You should see a folder labeled 'webapps'. Copy this folder to the Tomcat installation
  - Rename or delete the existing 'webapps' folder under the Tomcat install folder
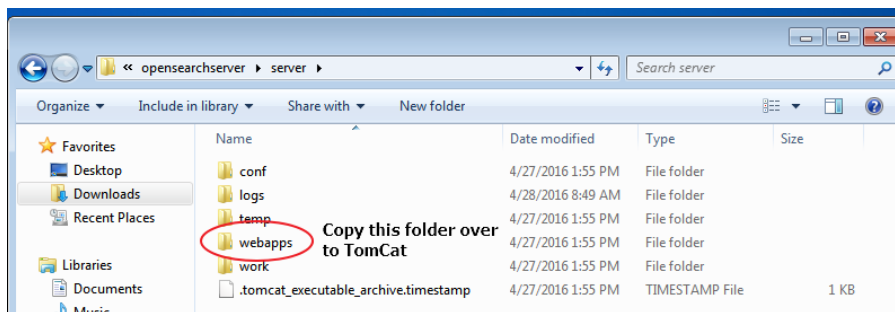
- Now we'll create a folder to store our indexes. If you don't know what those are, don't worry. We'll cover that later on. For now, just know that this folder is where all the information about what web pages have what words and phrases go. Feel free to name this folder to whatever you like.

- Create a system environment variable on your machine, called OPENSEARCHSERVER_DATA, pointing to the index folder you created in the last step.
- Open up Services from your Start menu. You should see Apache Tomcat listed as one of your services. Set it to run automatically and then start, or restart it if it's already running.
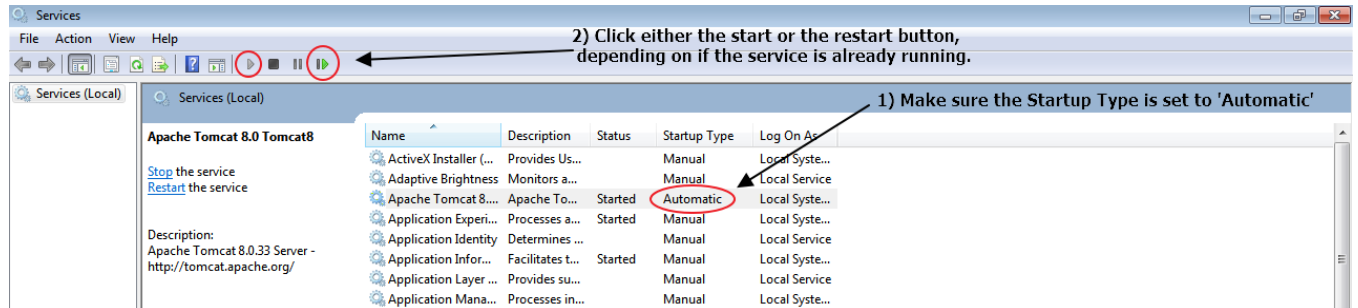

OSS is now running automatically on your machine without you having to run it from a batch file. Now it's time to configure it.

## *Configuration*

Connect to OSS and Create Accounts

Open a web browser, and navigate to your OpenSearchServer control panel. Since I'm running mine locally in this example, I'll go to http://localhost:9090/. You initially connect with a default, unauthenticated account. While you can configure and use OSS this way, we will create proper user accounts first.
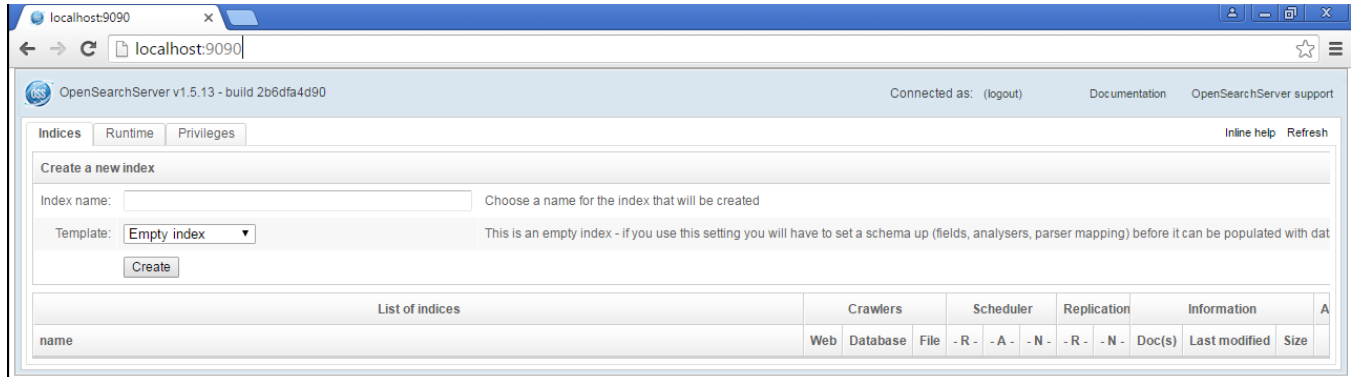
Next, we're going to add some user accounts. For now, we'll add two accounts: the Administrator account to manage OpenSearchServer, and the Searcher Account for Rock or any other web service to use while using the OSS API.

Open the privileges tab, and fill out the information below to create your administrator account. Once you're done with that, click the 'create' button to create the account. Don't worry about filling in the API Key field, one will automatically be generated for you.

- Administrator Account
    - o User name: administrator
    - o Password: <password>
    - o Admin user: CHECKED
    - o Can monitor: CHECKED
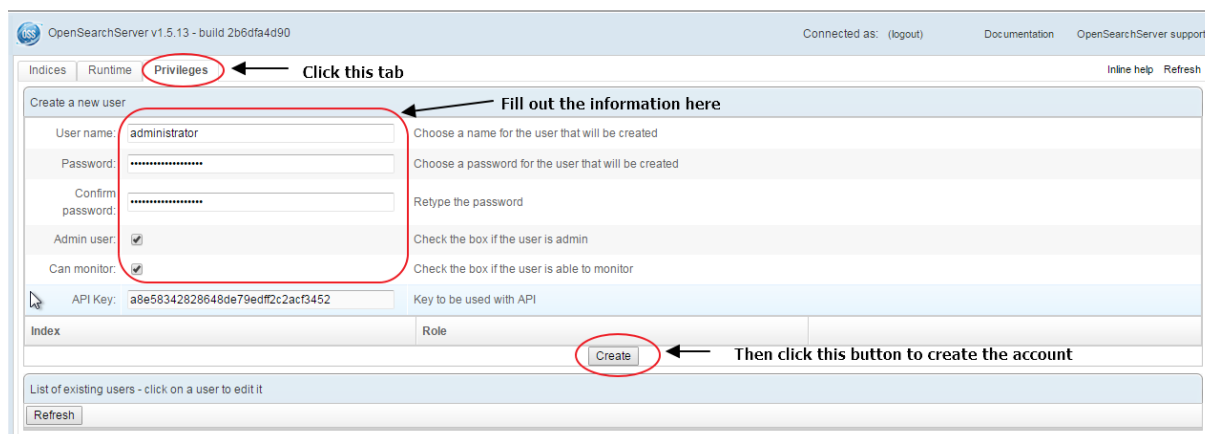
Repeat this process to create the Searcher account, using the information below:

- Searcher Account
  - User name: searcher
  - Password: <password>
 - Admin user: UNCHECKED
  - Can monitor: UNCHECKED

Now, logout of OpenSearchServer and log back in using the Administrator account you just created. Congratulations! You've successfully added the accounts you need to set up search on your website.
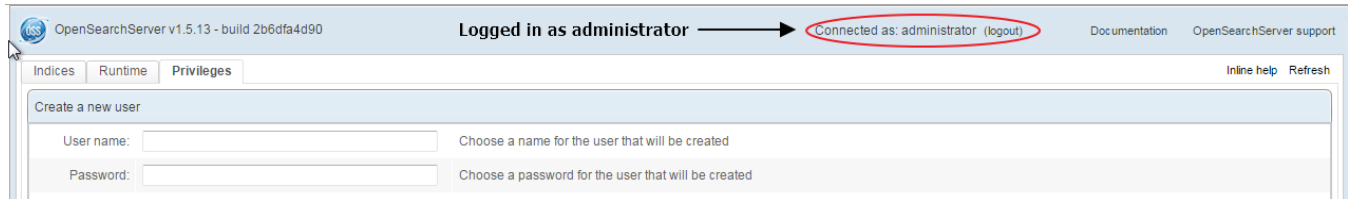

Creating the Index

In this section we'll be setting up our index and crawler. If you don't know what those are, the **crawler** is the part of the search engine that scans your websites for words and phrases, and the **index** is where the crawler stores all the information it comes across. When a user searches for a phrase, the website looks up in the **index** where any web pages that have that phrase exist and then returns the results.

First, we'll set up our index. Click the Indices tab, and a form to create a new index will appear. Enter 'RockIndex' for your index name, and make sure that the 'web crawler' template is selected. Once, that's done, click the 'create' button. You've now created the index that'll store all your web pages.
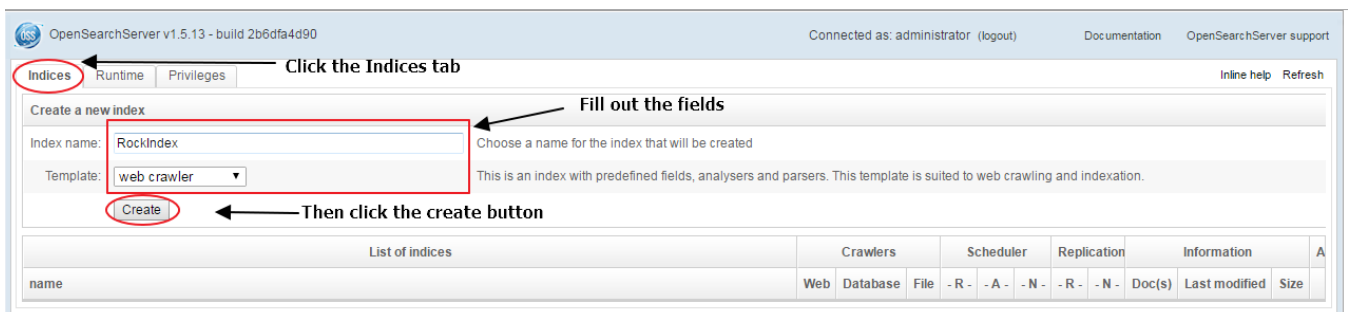
After you create your first index you'll see a lot more tabs pop up. We'll be using these to finish setting up OpenSearchServer for our website.
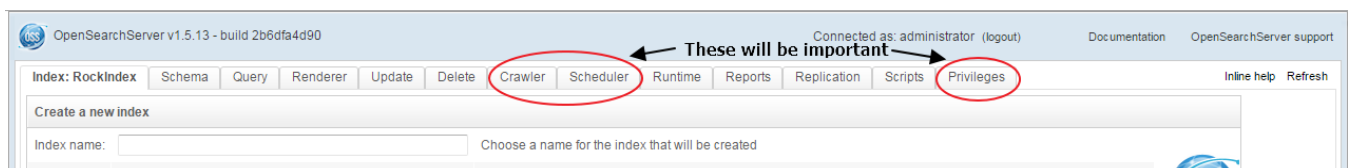
Configuring the Crawler

And now it's time to set up our crawler. Click the 'Crawler' tab, followed by the 'Web' tab. After that, set up your crawler like so:

On the Pattern List Tab:

- Check the 'Enabled' checkbox
- In the text box, enter the URL for your website, as per the second example given. For my test instance, I'm running my website at www.localrock.local, so I'll enter: http://www.localrock.local/*
- Click the 'Add' button

This tells the crawler which websites to search.
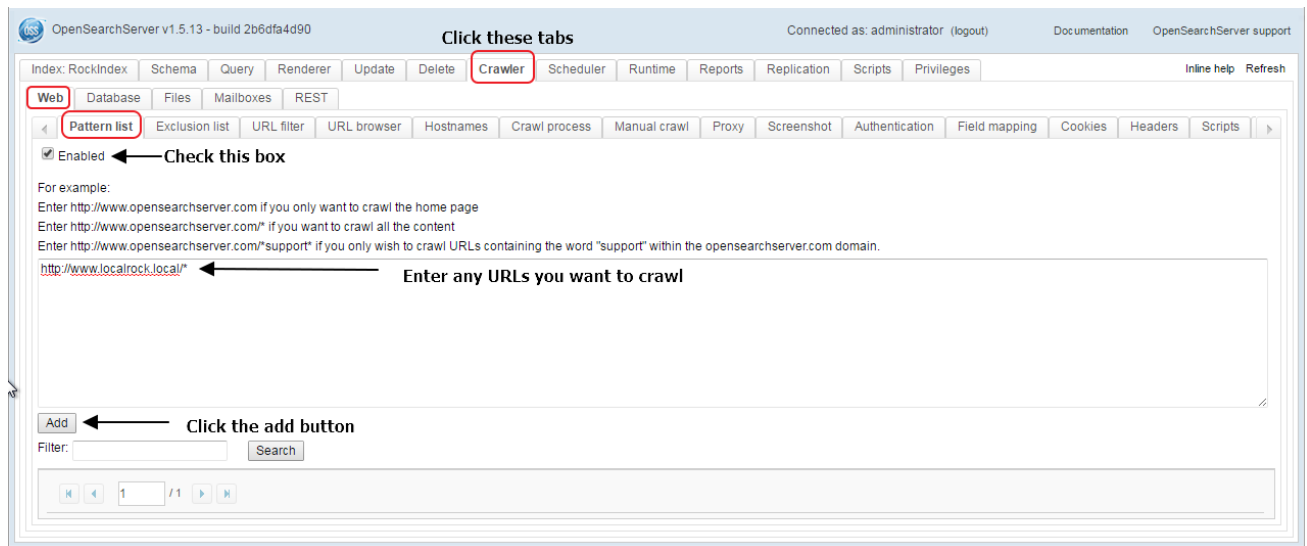
On the Exclusion List Tab:

Sometimes you may need to exclude pages and parts of your website from a crawl. For example, if you have sensitive content in a series of pages, you wouldn't want those to pop up in a search. To exclude them, do the following:

- Check the 'Enabled' checkbox
- In the text box, enter the URL(s) for page(s) you want excluded from your crawl. For my test instance, I'm running my website at www.localrock.local, so I'll enter: http://www.localrock.local/sensitivecontent
- Click the 'Add' button

Those pages are now excluded from the crawler's search, and won't pop up in any searches.
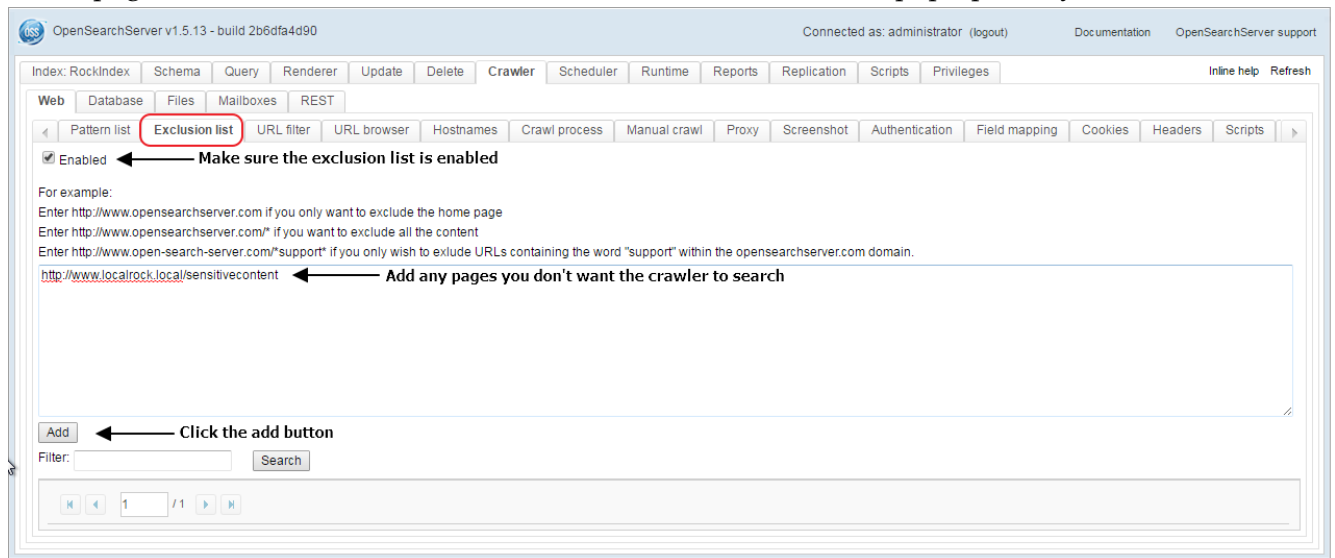
NOTE: In addition to sensitive content, you may want to exclude individual pieces of your website. For example, in our navigation bar, we have a link titled 'Events' that directs the user to our events page. Since our nav bar exists on every page in our website, when a user searches for 'events', every page will be returned as a result. To prevent this, you can add a div that excludes any content within it from being crawled: <div class="opensearchserver.ignore">Do not index!</div>

When you do this, OpenSearchServer will ignore everything within the div, neither indexing them to be used for search results nor traversing any links inside. So if you do this on a nav bar, be sure to also have a sitemap page somewhere else on your website for the crawler to traverse. For more information about this, check out this link:

http://www.opensearchserver.com/documentation/faq/crawling/how_to_exclude_some_part_of_webpage_from_crawling.md

On the Robots.txt Tab

- OSS supports a robots.txt file. If you have a properly formatted robots.txt and want it to govern OSS, leave the Enabled checkbox CHECKED. Otherwise, UNCHECK this to allow OSS to ignore your robots.txt file.
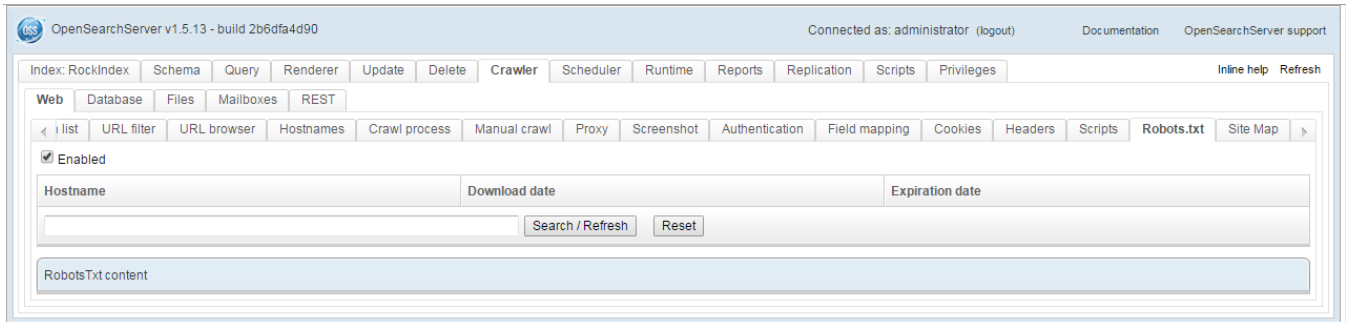
Scheduling the Crawler

Now that we've set up our crawler, it's time to set a schedule for it. At the top of the screen, you'll see a Scheduler tab. Click it, and you'll be taken to a page where you can manage the schedules for your crawler. Since we don't have any crawler jobs yet, we'll click the 'Create new scheduler job' button to create a new one.
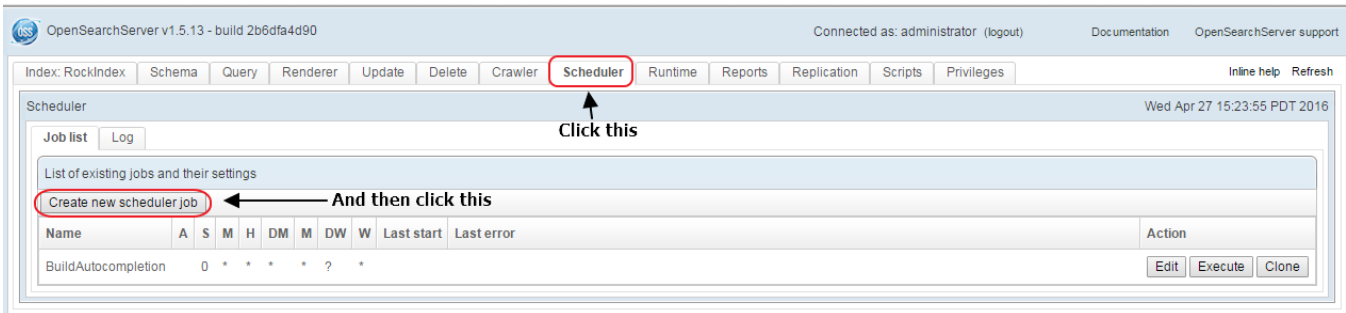
Once you're on the New Job page, fill out the following information:
- Job name: Crawl Website
- Active: CHECKED (enabled)
- Cron expression (When the job should fire)
    - o Click the Documentation button for info on cron jobs, if you are not familiar.
    - o Some example settings:
        - Run every 30 minutes: 0 0/30 * * * ? *
        - Run every hour: 0 0 * * * ? *
        - Run daily at 11:00pm: 0 0 23 * * ? *
        - Run every Friday at midnight: 0 0 0 FRI * ? *
- In the 'Tasks' section, set the task to be 'Web crawler – start'
    - o Leave Properties Run Once value = true. Otherwise this job will just run into infinity.
    - o Click the PLUS button under action to add the task

Once you're done entering the information, click the 'Create' button at the bottom.

You'll now be taken back to the job list page. Go ahead and click the 'Execute' button on your new job, and it'll run the next time it's scheduled for.

Congratulations! You've successfully configured your search engine! It'll now start to crawl your website next time it runs.

Monitoring the Index and Crawler

Monitor Active Crawls

If you want to monitor your active crawls, go to the Crawler -> Web -> Crawl Process Tab

- Click refresh in upper-right corner of screen
    - o Crawl details will be in the 'Statistics for prior sessions' area

To search your index

- Renderer Tab
    - o On 'default' renderer line, click VIEW button.
        - ▪ This will open a browser window with a search box.

Notes

- It will likely take multiple crawl sessions for OSS to visit and index every page on your site. Running as described above, you will likely see new URLs added to the index with each scheduled session run. To speed things up, you can manually trigger a RunForever crawl.
  - Crawler -> Web -> Crawl Process Tab
    - In the 'Current Status' area, select RunForever in the dropdown.
    - Click the button 'NOT RUNNING - CLICK TO RUN'.
  - This will run crawls one right after the other. You can monitor these until you don't see large numbers of new URLs being added with every session. At that point, you can click the 'RUNNING - CLICK TO STOP' button and let the scheduler take over.

Using the Rock Block to Add Search to Your Website

Configure the 'Searcher' User Account

Now it's time to set up our Rock Block. First, we'll need to grant the 'searcher' account we created the ability to query the index. Navigate to the Privileges tab and do the following:

- Select 'searcher' account at bottom of screen, under List of existing users
- In section, "Add privileges on following index":
  - Select 'RockIndex'
  - Select 'Index: query the index'
- Click the 'Add' button, and then the 'Save' button to save your changes.

NOTE: The API Key listed for this user will be used in the Rock Block configuration. Don't forget to keep it handy!


Windows Firewall

We'll also need to allow incoming traffic to OpenSearchServer in order for Rock to access it. Add a rule to the Windows Firewall on the machine running OSS to allow inbound traffic to port 9090 (the default), or whatever port you assign to the OSS application.

Install and Configure the Rock Block

The Search Block sends a JSON search query, using the Page Parameter 'SearchTerm' as the search term, to the OSS Server. It then passes the search results into Lava for you to display as you like. By using this method, you can pass in a search term while navigating to the page with the block on it, rather than having to type in your search term after loading the page.

If you have this document, you already have the Search Block in Rock. While you can add it to any page you like, an example has been provided under Admin Tools -> Power Tools -> OSS Search:
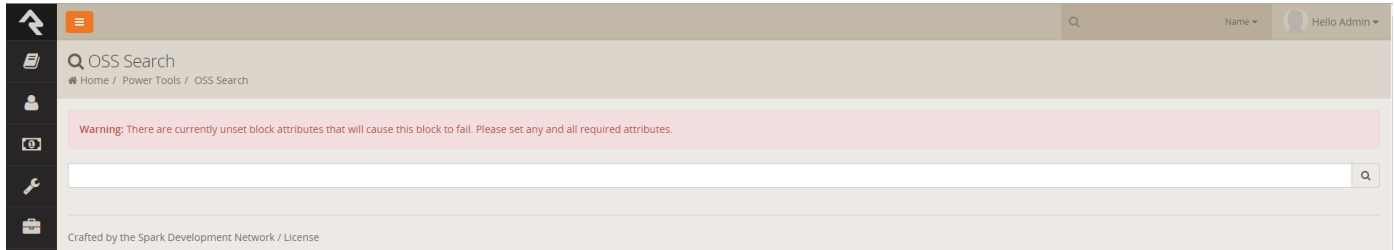
Before you get it up and running, however, you'll need to set some things in the block settings:

- **IP Address**: This is the IP address and port number that your OSS server is located at. Example: xx.xxx.x.xxx:9090
- **Index Name**: This is the name of the index that the server will search. By default, it's set to 'RockIndex', but if you named your index something else you'll need to enter it in here.
- **Searcher Username**: This is the username of the 'Searcher' user account OpenSearchServer uses to authorize an index query. By default, it's set to 'searcher'. If you named your searcher account something else, enter the correct name here.
- **API Key**: The API key of the 'Searcher' user account OpenSearchServer uses to authorize an index query. When you added privileges to the searcher account in the last step, there was a field for the account's API Key. Copy that value into this field.
- **Search Query**: This is the JSON search query that is sent to the server. The default query provided out-of-box can be found here: http://www.opensearchserver.com/documentation/api_v2/searching_using_fields/search.md If you want to customize your search query, like adding more or less weight to page titles, this is an excellent page that explains what each parameter in the query does, as well as how to modify it: http://www.opensearchserver.com/documentation/api_v2/search_parameters/README.md
- **Lava template**: This is the lava template provided to display the search results. Feel free to customize it to your heart's content!

Block Properties

Basic Settings   Advanced Settings

Name *

Search Results Lava

IP Address ❓ *   Enter the IP Address of your OpenSearchServer here. For example: xx.xxx.x.xxx:9090

Index Name ❓ *

RockIndex

Searcher Username ❓ *

searcher

Api Key ❓ *   Paste the API Key for the searcher account here

Search Query ❓

```
1   {
2       "query": "QUERY",      ◄── This is where the search term gets pasted into the query before being sent to OSS
3       "start": 0,
4       "rows": 10,            ◄── This dictates the number of results returned
5       "lang": "ENGLISH",
6       "operator": "AND",     ◄── This is where you can specify search logic. By default, it searches for matches to all search words, so it's set to AND.
7       "collapsing": {             If you want to return results that match any of the search words, set this to OR.
8           "max": 2,
9           "mode": "OFF",
10          "type": "OPTIMIZED"
11      },
12      "returnedFields": [
13          "url"
14      ],
15      "snippets": [
16          {
17              "field": "title",
18              "tag": "em",
19              "separator": "...",
20              "maxSize": 200,
21              "maxNumber": 1,
22              "fragmenter": "NO"
23          },
24          {
25              "field": "content",
26              "tag": "em",
27              "separator": "...",
28              "maxSize": 200,
```

Lava Template ❓ *

```
1   {% include '~/Plugins/com_centralaz/OpenSearchServer/Assets/Lava/SearchResultList.lava' %}
```
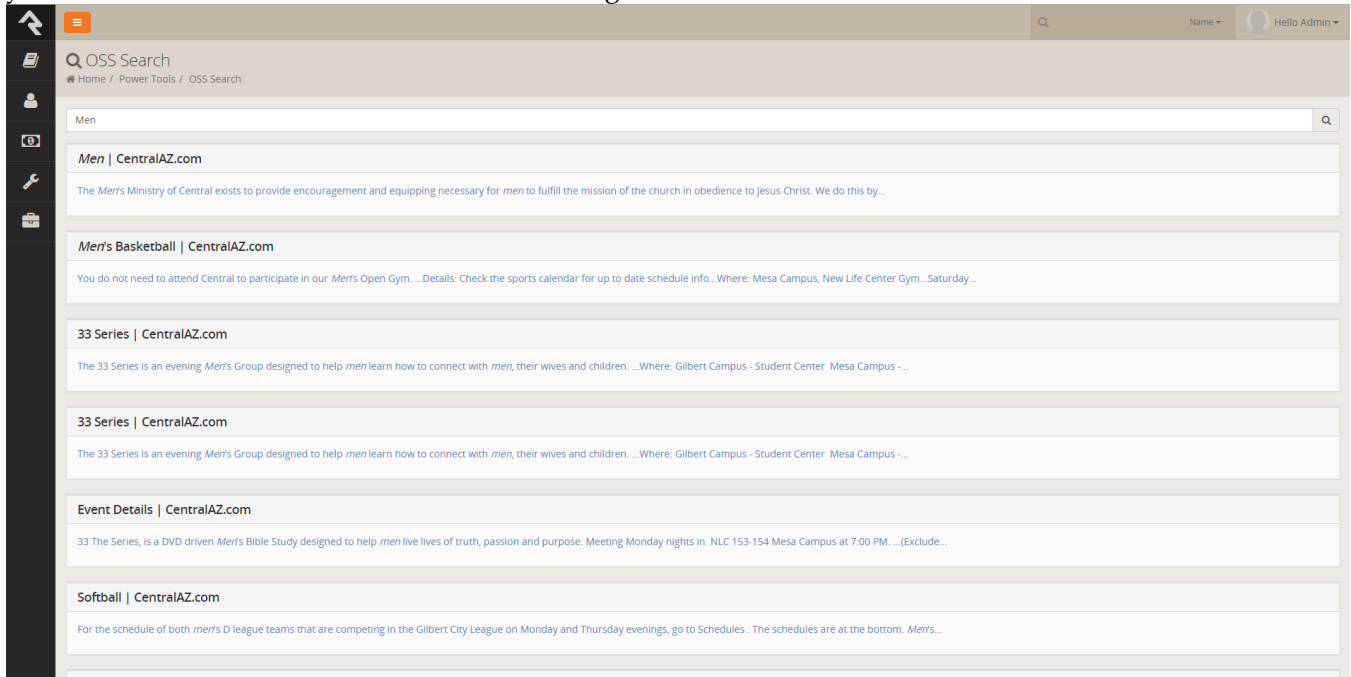
Enable Debug ❓

No ▾

Save   Cancel

Once you've filled out all these settings, click the 'Save' button. Voila! You now have search running on your website. Go ahead and search for something!

Wrap Up

You have website search capabilities now! Go knock it out of the park!

Resources

- RockRMS website
  - http://www.rockrms.com/
- OpenSearchServer website
  - http://www.opensearchserver.com/
- OSS online documentation
  - http://www.opensearchserver.com/documentation/README.md
- OSS API v2 Docs
  - http://www.opensearchserver.com/documentation/api_v2/searching_using_fields/search.md
  - http://www.opensearchserver.com/documentation/api_v2/search_parameters/README.md
- Run OSS as a Service
  - http://sourceforge.net/p/opensearchserve/discussion/947148/thread/9178569b/
  - https://tomcat.apache.org/tomcat-8.0-doc/windows-service-howto.html#Tomcat_service_application