Add-ons

Introduction

Add-ons for HUBzero are available here. Currently these consist of a couple projects that have not yet been fully integrated into the the HUBzero packaging and installation process.

Solr-powered Search

Introduction

Apache Solr is a search engine platform which is relatively mature and has a lot of powerful and flexible configurations. There has been extensive work to implement it into the HUBzero CMS and is currently a work-in-progress.

Solr is an open-source, mature, and stable searching service that is built upon the Apache Lucene search engine. The service provides features which lend itself to scaling and has a rich open source community. It is a Java-based service which provides search results through HTTP. Many companies such as Instagram, eBay, and StubHub rely on Solr to provide advanced searching capabilities.

The integration with Solr is currently under heavy development. It is **strongly recommended** to test on a QA / Stage host before using in a production environment.

Installation & First Time Configuration

Step 1: Install the hubzero-solr package

A system administrator must install the hubzero-solr RedHat or Debian Package using a package manager such as yum or aptitude. The package contains a version of Apache Solr and the configuration necessary for Solr to integrate with the CMS.

```
For RedHat / CentOS:
$    sudo yum install hubzero-solr
```

```
For Debian:
$    sudo apt-get install hubzero-solr
```

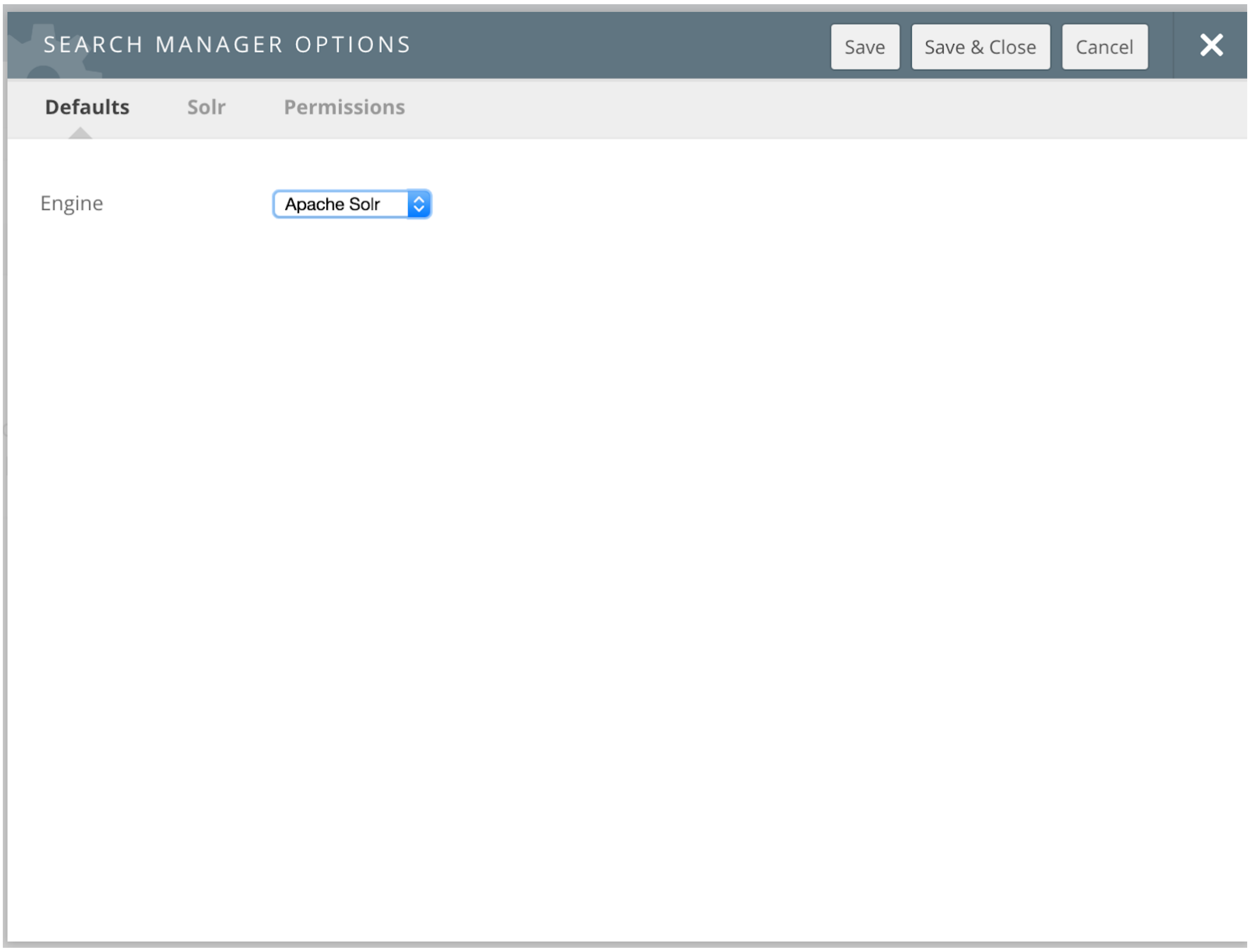
Once installed the service will need to be enabled.

```
$    sudo service hubzero-solr start
```

Step 2: Configure Search Service in the CMS

The HUBzero CMS needs to know to use Solr Search instead of Basic Search. To do this, a Hub administrator will need to log into the Administrative Backend and Configure the Search Component.

You will need to set **Engine** to *Apache Solr.* Then click the "Solr tab".



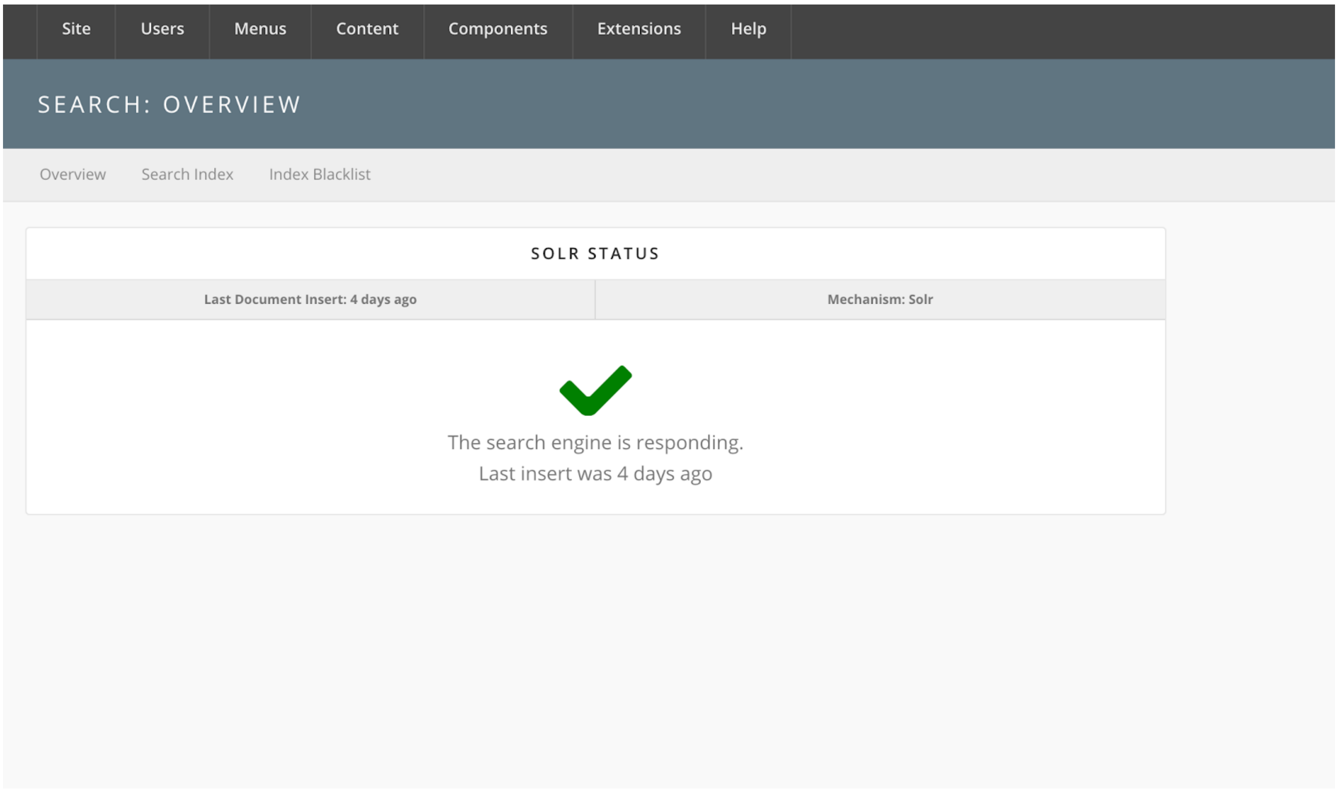
Engine: Apache Solr

The Solr tab's default settings will work for the open-source distribution.

HUBzero-hosted hubs are configured with different ports! The following scheme is used:

```
Development (dev.hub.org): 2090
Stage (stage.hub.org): 2091
Scan / QA (qa.hub.org): 2092
Production (hub.org): 2093
```

Click "Save and Close" to save the settings. If the hubzero-solr service is started and the correct settings were set in the steps above, the status screen should indicate that the search engine is responding.



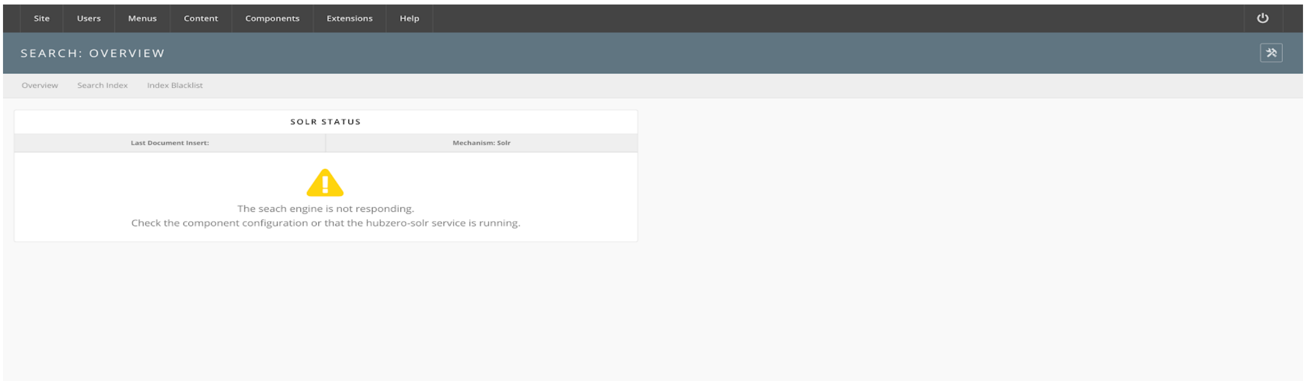
SOLR STATUS

Last Document Insert: 4 days ago

Mechanism: Solr

The search engine is responding.
Last insert was 4 days ago

If there were any issues with configuration, the following screen will appear.



SOLR STATUS

Last Document Insert:

Mechanism: Solr

The seach engine is not responding.
Check the component configuration or that the hubzero-solr service is running.

This would be a point where a support ticket is filed for the system administrator to confirm that the service is running. Please include all configuration parameters contained in Step #3 when filing the ticket.

Step 3: Enable the Search Background Worker

In order to keep the search index fresh, a background worker is implemented to process data from the CMS and push it into the Solr service.

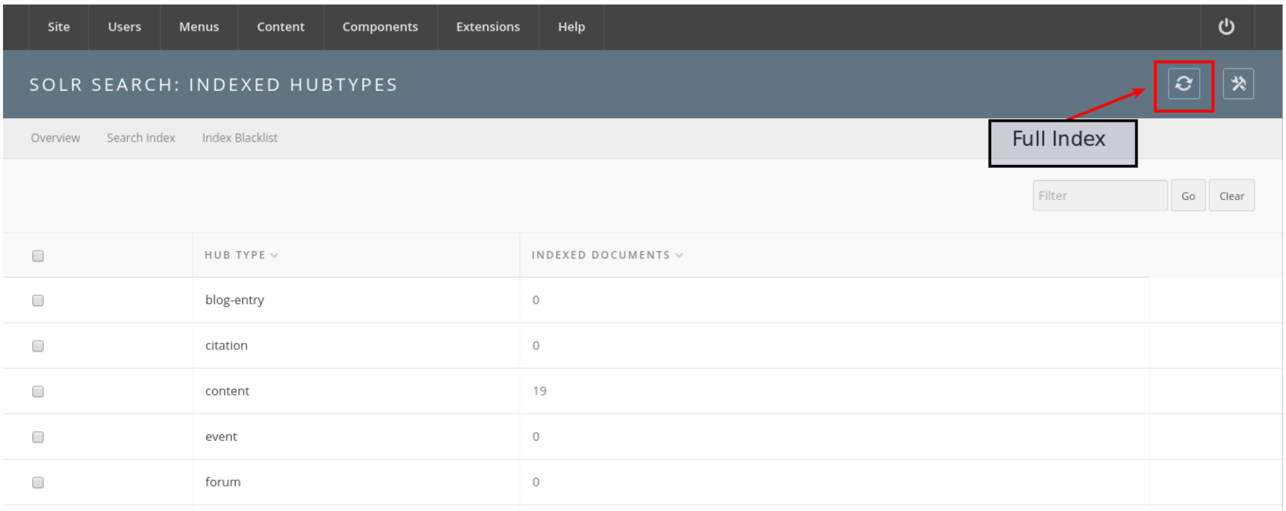
Currently the background worker is implemented as a Cron task that is called once a minute. There is work being done to develop a daemon which listens to CMS events and processes data without relying on Cron.

To setup the Cron-based worker a Hub administrator must go into the Administrative Backend, go to Components, Cron, and add the Task as shown below:

Click "Save and Close".

Step 4: Build the Initial Index

This implementation of Solr has hooks into the CMS which updates the index when a new record is added or marked for deletion. It will be necessary to add items which have been added before Solr was activated.

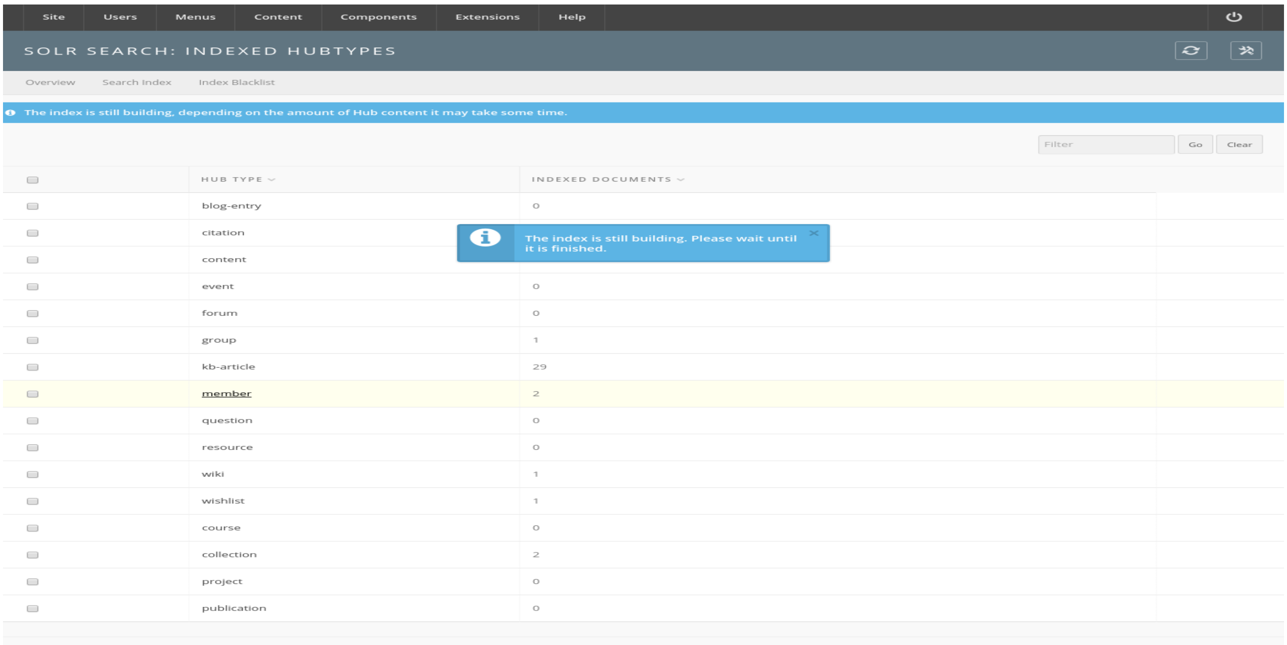


SOLR SEARCH: INDEXED HUBTYPES

Overview · Search Index · Index Blacklist

☐	HUB TYPE	INDEXED DOCUMENTS
☐	blog-entry	0
☐	citation	0
☐	content	19
☐	event	0
☐	forum	0

This operation should only need to be completed once. You will be unable to start this operation until it finishes for the first time.



SOLR SEARCH: INDEXED HUBTYPES

Overview · Search Index · Index Blacklist

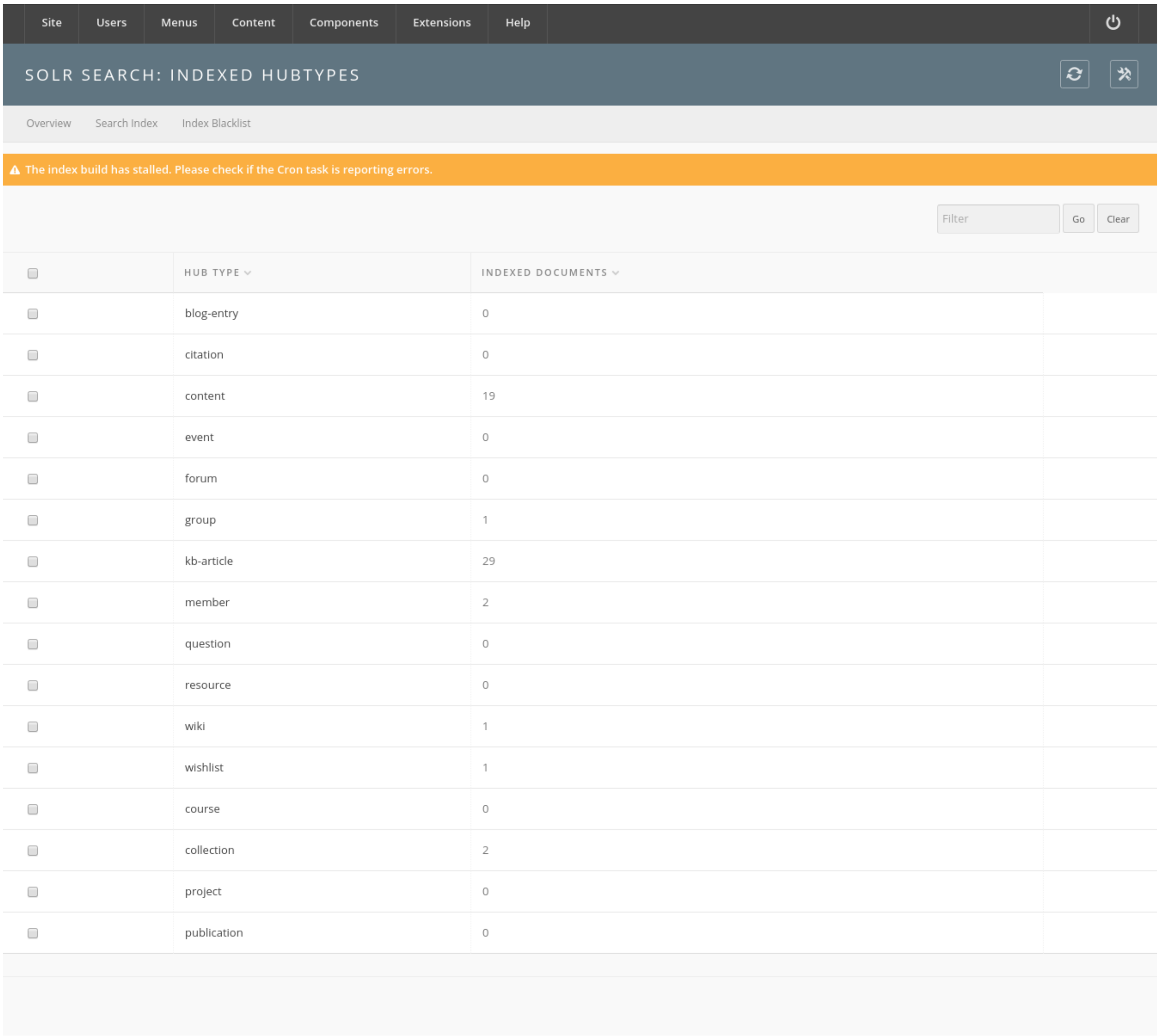
The index is still building, depending on the amount of Hub content it may take some time.

The index is still building. Please wait until it is finished.

☐	HUB TYPE	INDEXED DOCUMENTS
☐	blog-entry	0
☐	citation	
☐	content	
☐	event	0
☐	forum	0
☐	group	1
☐	kb-article	29
☐	member	2
☐	question	0
☐	resource	0
☐	wiki	1
☐	wishlist	1
☐	course	0
☐	collection	2
☐	project	0
☐	publication	0

The "Full Index" button populates a Queue which is periodically serviced by a worker. The worker will process the records and format for consumption by the Solr service. **This may take several hours to fully complete if the Hub has a lot of content.**

If an error with the worker occurs, a warning message such as this will appear.



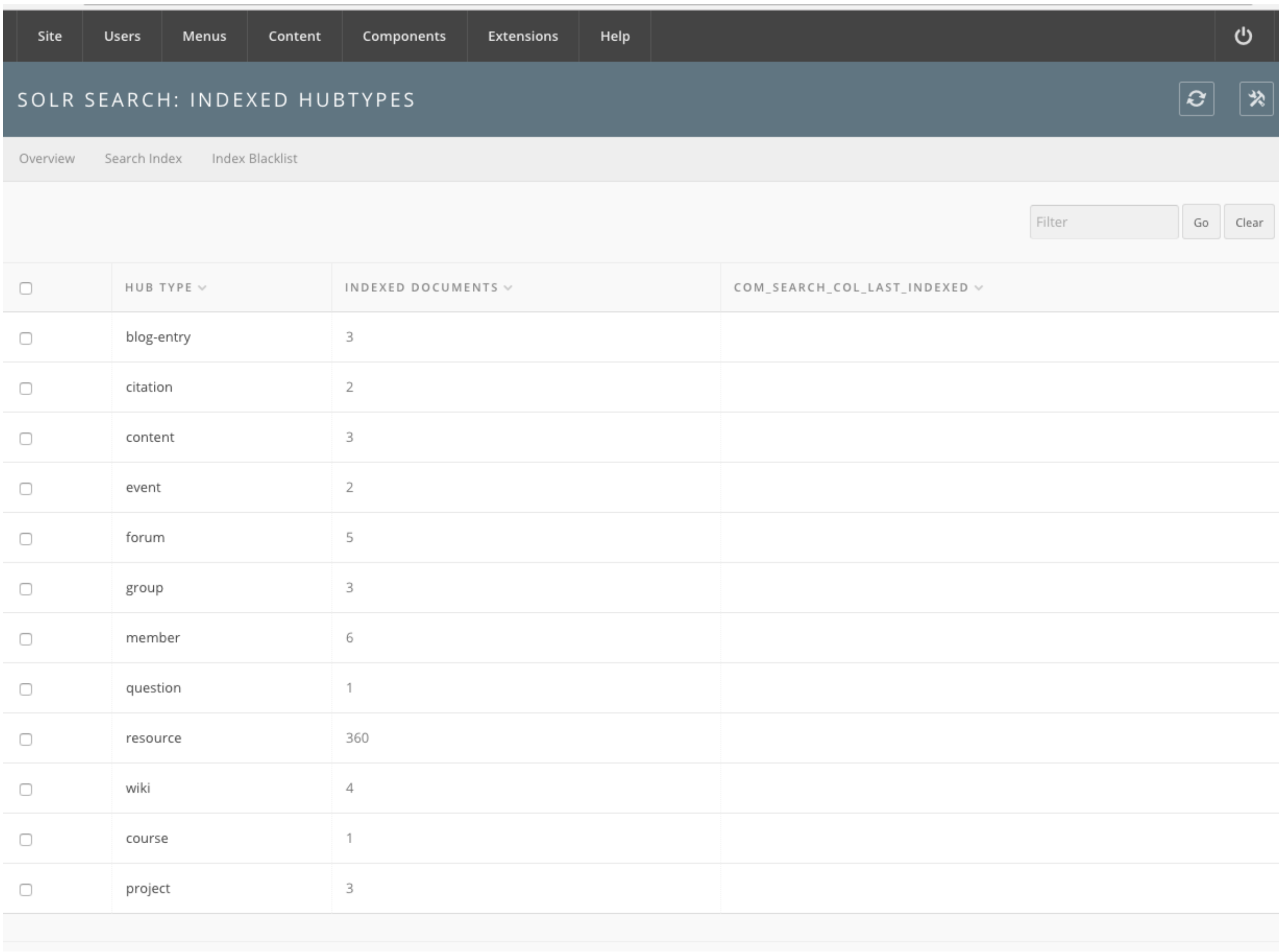
The index build has stalled. Please check if the Cron task is reporting errors.

	HUB TYPE	INDEXED DOCUMENTS
□	blog-entry	0
□	citation	0
□	content	19
□	event	0
□	forum	0
□	group	1
□	kb-article	29
□	member	2
□	question	0
□	resource	0
□	wiki	1
□	wishlist	1
□	course	0
□	collection	2
□	project	0
□	publication	0

Search Breadth

Te question is “What can I search for?”. The answer is “anything you have access to contained within the list in Search Categories. To see all content within these categories perform a simple query using the wildcard character “*” as shown below.

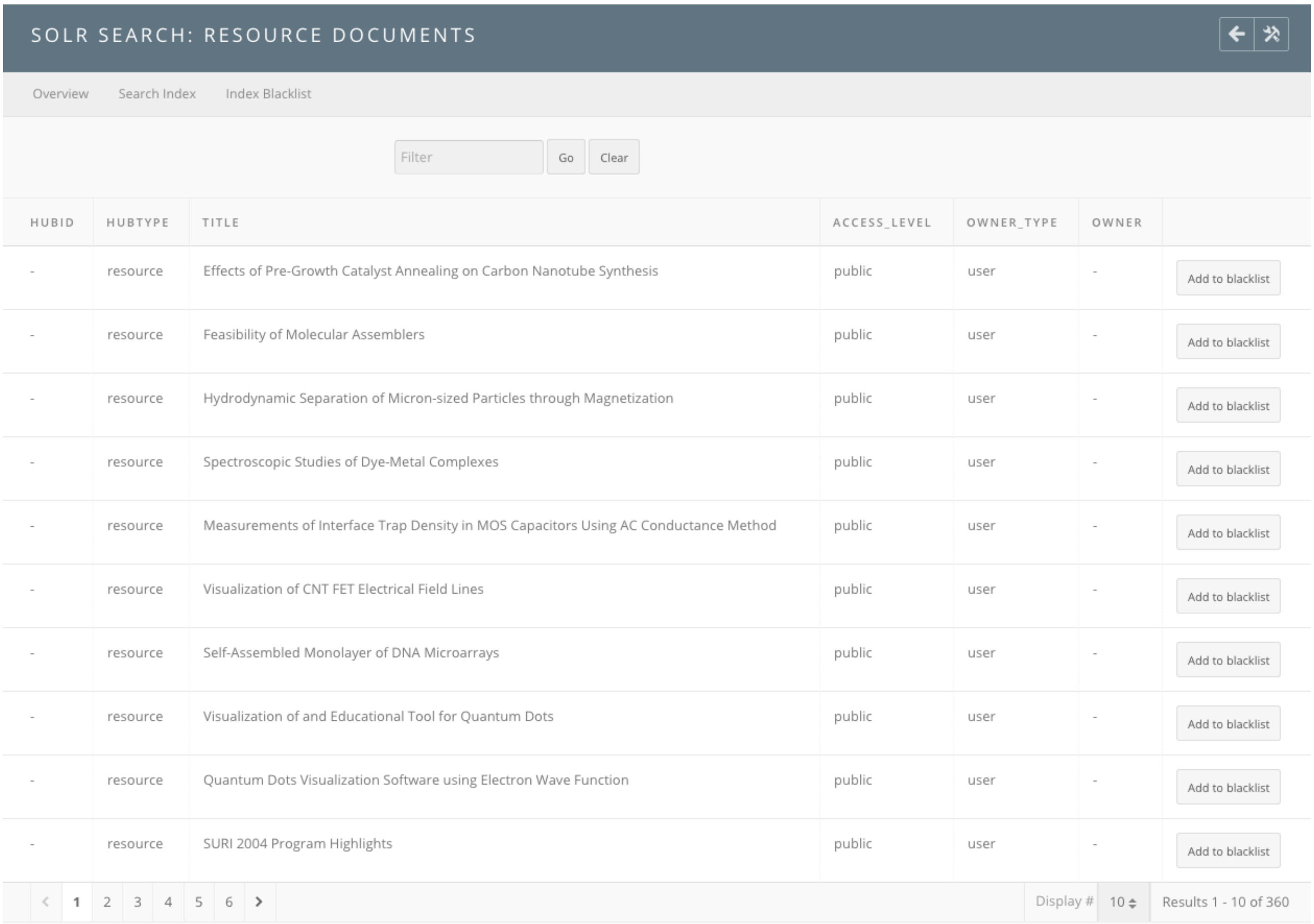
A better of what is currently inside Solr’s index can be viewed on the administrative backend by going to Components >> Search >> Search Index Tab. The number of index items is located to next to each type. Clicking on the name of the hub type will perform a search on that type, displaying all items that are within the index of that type.



SOLR SEARCH: INDEXED HUBTYPES

□	HUB TYPE	INDEXED DOCUMENTS	COM_SEARCH_COL_LAST_INDEXED
□	blog-entry	3	
□	citation	2	
□	content	3	
□	event	2	
□	forum	5	
□	group	3	
□	member	6	
□	question	1	
□	resource	360	
□	wiki	4	
□	course	1	
□	project	3	

For instance clicking “Resources” shows the following screen:



HUBID	HUBTYPE	TITLE	ACCESS_LEVEL	OWNER_TYPE	OWNER	
-	resource	Effects of Pre-Growth Catalyst Annealing on Carbon Nanotube Synthesis	public	user	-	Add to blacklist
-	resource	Feasibility of Molecular Assemblers	public	user	-	Add to blacklist
-	resource	Hydrodynamic Separation of Micron-sized Particles through Magnetization	public	user	-	Add to blacklist
-	resource	Spectroscopic Studies of Dye-Metal Complexes	public	user	-	Add to blacklist
-	resource	Measurements of Interface Trap Density in MOS Capacitors Using AC Conductance Method	public	user	-	Add to blacklist
-	resource	Visualization of CNT FET Electrical Field Lines	public	user	-	Add to blacklist
-	resource	Self-Assembled Monolayer of DNA Microarrays	public	user	-	Add to blacklist
-	resource	Visualization of and Educational Tool for Quantum Dots	public	user	-	Add to blacklist
-	resource	Quantum Dots Visualization Software using Electron Wave Function	public	user	-	Add to blacklist
-	resource	SURI 2004 Program Highlights	public	user	-	Add to blacklist

Results 1 - 10 of 360

One can perform additional searching using the “Filter” bar on top of the results listing.

Shibboleth authentication

Shibboleth Authentication is an unsupported feature of HUBzero. This documentation has not been verified against HUBzero 2.2 or RedHat. It is included here for completeness. Please send any corrections or feedback to support@hubzero.org

Installation

If installing on Debian GNU/Linux

```
# apt-get install hubzero-shibboleth
```

If installing on Redhat/CentOS Linux

```
# yum install hubzero-shibboleth
```

If you installed the hubzero-shibboleth package on Debian, you're set. The relevant packages were included as dependencies. The packages are:

shibboleth2-sp-utils shibboleth2-sp-schemas libapache2-mod-shib2

At the time of this writing, Shibboleth is not distributed in the core repositories for Redhat/CentOS. You can read about how to add a repo that has what you need here:

https://wiki.shibboleth.net/confluence/display/SHIB2/NativeSPLinuxRPMInstall

Generate a private key

Use shib-keygen to generate /etc/shibboleth/sp-key.pem. Note that this utility may not be on your path unless you are root.

Configure Shibboleth

Shibboleth official quickstart documentation reference

The main configuration file is located at /etc/shibboleth/shibboleth2.xml. There are some other files that might be of interest to you here, but the defaults are acceptable to get your hub working with InCommon.

In shibboleth2.xml:

- Update <ApplicationDefaults entityID="{url}" ...> so {url} is https://{your hostname}/Shibboleth.sso. This is your Shibboleth endpoint, designated later by the Apache configuration as the location where the shib2 module will manage communication with ID providers.
- Update <Sessions ...handlerSSL="false" ...> to handlerSSL="true", if it is not already

Configure Apache

Shibboleth official Apache configuration reference

Ensure that Apache is loading the module. Typically this means that there is a link in mods-enabled to shib2.load in mods-available

ln -s /etc/apache2/mods-available/shib2.load /etc/apache2/mods-enabled

If you do not have this directory structure you can also enable the module directly in the next step by adding this to your Apache configuration file:

LoadModule mod_shib /usr/lib/apache2/modules/mod_shib2.so

In the conf file defining your SSL host, (usually located in /etc/apache2/sites-enabled):

- If not already set in the SSL <VirtualHost> UseCanonicalName on;
- To enable shibd's endpoint, add: <Location /Shibboleth.sso> SetHandler shib </Location>

- HUBzero CMS routing will stomp on /Shibboleth.sso unless you change the mod_rewrite rules a bit.
    - You should have a line like: RewriteRule (.*) index.php probably preceded by a few 'RewriteCond's. Add a new condition to exempt the shib2-controlled path:

```
RewriteCond %{REQUEST_URI} !/Shibboleth.sso/.*$ [NC]
```

Restart apache: /etc/init.d/apache2 restart

Verify

From the same host (this is IP-restricted):

wget -q --no-check-certificate https://localhost/Shibboleth.sso/Metadata -O - | tee /etc/shibboleth/sp-metadata.xml

This command should write XML to the listed file (and stdout) wrapped in <md:EntityDescriptor xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata" ...>

If it does not, review the references above to troubleshoot.

You may skip to "Configuring HUBzero CMS!" if you do not want to test interop more thoroughly with the TestShib ID provider, but I would recommend you do this test.

Upload metadata to TestShib.org

- Copy the metadata generated above to some unique name, for example:

cp /etc/shibboleth/sp-metadata ~/{your hostname}-sp-metadata.xml

- Upload that file here: https://www.testshib.org/register.html. Uploading a file of the same name will overwrite it on the testshib server, should you need to make any adjustments.

Change your local configuration to accept TestShib as an ID provider

- Visit this URL to get an appropriate test configuration XML file:

https://www.testshib.org/cgi-bin/sp2config.cgi?dist=Others&hostname={your hostname from the Shibboleth Configuration step above})

- Assuming that looks OK, copy the output over your existing

```
/etc/shibboleth/shibboleth2.xml:

wget -q --no-check-certificate "https://www.testshib.org/cgi-bin/sp2config.cgi?dist=Others&hostname={your hostname}" -O /etc/shibboleth/shibboleth2.xml
```

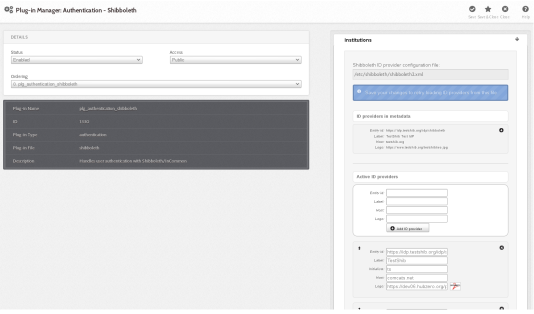
Restart services: /etc/init.d/shibd restart && /etc/init.d/apache2 restart

Configuring HUBzero CMS

If you do not already have plg_authentication_shibboleth installed, this package installs a tarball in $(PREFIX)/usr/lib/hubzero that you may install using HUBzero CMS's package management interface at /administrator.

Manage ID providers on HUBzero CMS's admin page

In Extensions->Plugins, select Authentication - Shibboleth



Plug-in Manager: Authentication - Shibboleth

DETAILS

Status: Enabled

Access: Public

Ordering: 0. plg_authentication_shibboleth

Plug-in Name	plg_authentication_shibboleth
ID	1330
Plug-in Type	authentication
Plug-in File	shibboleth
Description	Handles user authentication with Shibboleth/InCommon

Institutions

Shibboleth ID provider configuration file: /etc/shibboleth/shibboleth2.xml

Save your changes to retry loading ID providers from this file

ID providers in metadata

Entity id: https://idp.testshib.org/idp/shibboleth
Label: TestShib Test IdP
Host: testshib.org
Logo: https://www.testshib.org/testshibtwo.jpg

Active ID providers

Entity id:
Label:
Host:
Logo:

Entity id: https://idp.testshib.org/idp/s
Label: TestShib
Initialism: ts
Host: comcats.net
Logo: https://dev06.hubzero.org/p

Ideally it should look a lot like the screenshot in that it found testshib in your XML configuration. If so, you can click the down arrow by that entry to move it into your active provider list.

This may fail if, for example, shibboleth2.xml is not readable by the web user, or if you changed your configuration so that the file is located somewhere unexpected.

It is not necessary, however, for the web server to read this file. If you'd like you can simply enter the EntityID for testshib (https://idp.testshib.org/idp/shibboleth) in the white box with the button labeled "Add ID provider". Enter something, eg "TestShib" for the label.

Quick run-down of the fields here:

- Entity id: (required) corresponds to the corresponding entityId in shibboleth2.xml and must match exactly for things to work out.
- Label: (required) name to show on the log-in button of your hub for this provider
- Initialism: (optional) if you have more than ten supported ID providers, the log-in list becomes searchable, and in this case you can add a short name for institutions so that they will come up when the user types that as well as when they type a portion of the label. (For example, if you federated with the National Science Foundation you might add "NSF" here)
- Host: (optional) institutions may be pre-selected if the IP address of the user looks like it is in a particular network, eg, to follow the previous example, nsf.gov to pre-select the National Science Foundation
- Logo: (optional) also shown on the button. Enter a URL here to make a iconified copy of it. You may have better results in some cases if you resize to no more than 28px in either extent yourself.

Finally, you can select the order in which you would like the button to appear on the login page here. When you're done, click "Save & Close" in the top right. This will take you back to the screen where you can click the icon in the "Status" column to enable the plug-in.

Try logging in!

(If you run into any problems here, there might be a clue in TestShib's logs of its ID provider actions)

Since TestShib doesn't release any attributes, you'll have to enter a name when you log in. Hopefully you can negotiate to get names and emails released to your hub with "real" ID providers, which you're now clear to do if everything worked out.

Help?

- If you have a problem that you can't resolve that appears to be related to Shibboleth's machinery, please consult the official documentation carefully.

- If you can't resolve the problem, there is a mailing list: https://www.shibboleth.net/community/lists/

- If the problem you are experiencing appears to be related not to the Shibboleth interchange mechanism but to something in the hub's implementation of the log-in procedure, visit https://help.hubzero.org/support to enter a support ticket describing the situation.

- Test that you can access https://idp.purdue.edu/idp/shibboleth or similar directly.

InCommon

InCommon Authentication is an unsupported feature of HUBzero. This documentation has not been verified against HUBzero 2.2 or Debian 8+. It is included here for completeness. Please send any corrections or feedback to support@hubzero.org

Introduction

This plugin provides some code necessary to allow your hub to accept credentials using the Shibboleth system. Most commonly, this implies membership in the InCommon network.

Shibboleth has some particular architectural demands, namely that it will install a new daemon and a new Apache module on your system. InCommon has some administrative demands, in that you will need to negotiate to get your hub added to their XML manifest as a service provider.

Installation

Debian

```
# apt-get install -y libapache2-mod-shib2
```

Redhat Enterprise Linux & other distributions

See Shibboleth wiki entry on service provider installation for information on how to add the Shibboleth software to your list of repositories so that it can be installed and upgraded through yum, or, failing that, how to install from SRPMS.

Configuration

Shibboleth

Certificates

As root, run the script shib-keygen, which was installed as part of the package. This will generate a key pair for your service provider to use. No further configuration is required for this; the software will find the keys when the shibd service is restarted.

output

```
Generating a 2048 bit RSA private key
......................................................................
..................+++
.....................+++
writing new private key to '/etc/shibboleth/sp-key.pem'
-----
```

/etc/shibboleth/attribute-map.xml

This file controls which attributes (bits of user information) the software will extract during login when the identity provider makes them available.

Make sure the following pertinent attributes are not commented out in both forms of the “name” attribute.

eppn (username, probably already enabled in the shipped configuration):

```
    <Attribute name="urn:mace:dir:attribute-def:eduPersonPrincipalName" id="eppn">
        <AttributeDecoder xsi:type="ScopedAttributeDecoder"/>
    </Attribute>
    <Attribute name="urn:oid:1.3.6.1.4.1.5923.1.1.1.6" id="eppn">
        <AttributeDecoder xsi:type="ScopedAttributeDecoder"/>
    </Attribute>
```

Name & email (probably not enabled by default):

```
    <Attribute name="urn:mace:dir:attribute-def:sn" id="sn"/>
    <Attribute name="urn:mace:dir:attribute-def:givenName" id="givenName"/>
    <Attribute name="urn:mace:dir:attribute-def:displayName" id="displayName"/>
    <Attribute name="urn:mace:dir:attribute-def:mail" id="mail"/>
    <Attribute name="urn:oid:2.5.4.4" id="sn"/>
    <Attribute name="urn:oid:2.5.4.42" id="givenName"/>
    <Attribute name="urn:oid:2.16.840.1.113730.3.1.241" id="displayName"/>
    <Attribute name="urn:oid:0.9.2342.19200300.100.1.3" id="mail"/>
```

/etc/shibboleth/shibboleth2.xml

This is the main configuration, which controls how the software federates with identity providers.

First, replace $YOUR_HOSTNAME with, uh, your hostname, in the entityID attribute near the top of the file:

```
       <ApplicationDefaults entityID="https://$YOUR_HOSTNAME/login/shi
bboleth" REMOTE_USER="eppn persistent-id targeted-id">
```

In the block, delete or comment-out any SSO or SessionInitiator blocks that shipped, and add the two listed below, again interpolating your real hostname. This tells the software to check with the HUBzero CMS plugin about where to redirect for a given authentication request, and allows the HUBzero CMS to selectively enable providers.

```
        <Sessions lifetime="28800" timeout="3600" relayState="ss:mem"
                  checkAddress="true" handlerSSL="true" cookieProps="h
ttps">
            <SSO discoveryProtocol="SAMLDS" ECP="true" discoveryURL="h
ttps://$YOUR_HOSTNAME/login?authenticator=shibboleth&wayf">
                   SAML2 SAML1
            </SSO>
            <SessionInitiator type="Chaining" Location="/login/shibbol
eth" isDefault="true" id="Login">
                <SessionInitiator type="SAML2" template="bindingTempla
te.html"/>
                <SessionInitiator type="Shib1"/>
                <SessionInitiator type="SAMLDS" URL="https://$YOUR_HOS
TNAME/login?authenticator=shibboleth&wayf"/>
            </SessionInitiator>
            <!-- Default <Handler> tags not pictured, but they should
stay -->
       </Sessions>
```

If you run into issues where you seem to be stuck in a redirect loop between the idp and the sp, changing the cookie properties to use a less specific path may help.

```
cookieProps="; path=/; secure; HttpOnly"
```

If this is a production machine you will want to set a real email for the support contact:

```
       <Errors supportContact="support@$YOUR_HOSTNAME"
            helpLocation="/about.html"
            styleSheet="/shibboleth-sp/main.css"/>
```

Finally, you will need to configure how and where the software looks for metadata about identity providers. This is just a list of providers you can support, including some helpful annotations like where the service URLs and what public key to use when communicating with it.

Metadata provider: TestShib

For development and test machines it is often useful to use TestShib, and its configuration looks like this, below the Sessions tag and at the same scope:

```
      <MetadataProvider type="XML" uri="http://www.testshib.org/metada
ta/testshib-providers.xml" backingFilePath="testshib-two-idp-
metadata.xml" reloadInterval="180000"/>
```

Visit the TestShib site for more information about how to set this up, if you’re interested. Hopefully you do not need the “Install” selection, but pick up from “Register”. During “Configure” it recommends replacing your whole shibboleth2.xml with one it generated. Make a backup if you do, or else just add the MetadataProvider above to your existing configuration.

When you reach “Test”, see below for the HUBzero CMS configuration that will add TestShib to the list of available identity providers on your hub.

Metadata provider: InCommon

If your plans include membership in the InCommon consortium, this is the incantation, below the Sessions tag and at the same scope:

```
        <MetadataProvider type="XML" uri="https://wayf.incommonfederat
ion.org/InCommon/InCommon-metadata.xml" backingFilePath="federation-
metadata.xml" reloadInterval="7200">
            <MetadataFilter type="RequireValidUntil" maxValidityInterv
al="2419200"/>
            <MetadataFilter type="Signature" certificate="inc-md-
cert.pem"/>
        </MetadataProvider>
```

Install https://ds.incommon.org/certs/inc-md-cert.pem as /etc/shibboleth/inc-md-cert.pem so it’s available for this provider.

Metadata provider: others?

If you are doing one-on-one negotiations with identity providers the metadata situation gets a bit more hairy, but the identity providers in question will probably be able to guide your configuration.

Apache

Quoth the Shibboleth wiki entry on service provider installation:

- UseCanonicalName On
- Ensure that the ServerName directive is properly set, and that Apache is being started with SSL enabled.

Make sure installing the software enabled both the module shib2 and the support daemon shibd.

Typically this means that there is a symlink /etc/apache2/mods-enabled/shib2.load that points to /etc/apache2/mods-available/shib2.load and that this report works:

```
# service shibd status
[ ok ] shibd is running.
```

/etc/apache2/sites-enabled/{your-ssl-enabled-config-file}

Your EntityID is something like https://hostname/login/shibboleth, but the actual URL to pick up the login process again in HUBzero CMS terms is more complicated, so we rewrite it. I recommend putting this statement as high as possible in the config (after RewriteEngine on) so that the “L“ast last triggers and you can be assured the URL is not subsequently rewritten by anything else you’re doing.

```
        RewriteCond     %{REQUEST_URI}          ^/login/shibboleth
        RewriteRule     (.*) /index.php?option=com_users&authenticator
```

```
=shibboleth&task=user.login [NC,L]
```

Bind an endpoint to the module. This is used during the login process and is also useful to get a basis for your service provider’s metadata, which is served at /Shibboleth.sso/Metadata when the request comes from localhost.

```
        <Location /Shibboleth.sso>
                SetHandler shib
        </Location>
```

You probably have a rule that directs all requests that appear to be for HUBzero CMS content to the index.php bootstrap, and we need to note that /Shibboleth.sso isn’t HUBzero CMS business, so make sure you have a RewriteCond like this:

```
        RewriteCond     %{REQUEST_URI}          !^/Shibboleth.sso/.*$
[NC]
        RewriteRule     (.*)                    index.php
```

Finally, we actually protect the entityID location /login/shibboleth. We can redirect a user to this path to require them to make a Shibboleth login. Shibboleth won’t know specifically how to do that so it will make a request to the wayf location defined above in shibboleth2.xml. This is part of the HUBzero CMS that knows already which provider the user selected from the login page, so it spits back the appropriate identity provider entityId. From there the metadata is referenced to find the endpoint associated with that institution, and the user is sent to the login page. They come back to /login/shibboleth upon submission, but now the requirement to have a Shibboleth session is satisified, and the rewritten URL referencing user.login is served to complete the process.

```
        <Location /login/shibboleth>
                AuthType shibboleth
                ShibRequestSetting requireSession 1
                Require valid-user
        </Location>
```

Restart the shibd and apache2 services when satisified with this configuration.

HUBzero CMS

Plugin

Log in to /administrator, choose Extensions and then Plugin Manager, and locate the Shibboleth plugin in the Authentication category.

If you would like to selectively hide the Shibboleth login options for testing, enter something in the “Testing mode key” field. This term must appear in the query string for the controls of the plugin to render. For example, if you enter “incommon” you should test the login page by visiting “/login?reset=1&incommon” (reset=1 in case it remembers your logging in with a different method, in which case you’ll only see the controls for that method anyway).

The links section is useful only for testing, but it can be used to destroy a link between your test account and a particular institution so that you can try it again.

The Institutions section is where you manage which options are presented on the login page.

An example of an entry here, for the TestShib public identity provider test mechanism described above:

```
Entity ID: https://idp.testshib.org/idp/shibboleth
Label: TestShib
Host: testshib.org
```

The entity ID must strictly match what you have in your metadata provider, but the label is free-form and the host is optional. The login page attempts to do a reverse-DNS of the user’s IP to see if they are on a particular network. If it turned out in this case that the client was from *.testshib.org this option would be pre-selected in the plugin’s controls.

Save your settings with the button near the top right of the page when you’re done.