



MUSE

MuseKnowledge™ Proxy Integration

MuseGlobal, Inc.
421 Fayetteville Street
Suite 1100
Raleigh, NC 27601,
U.S.A.
www.museglobal.com

MuseGlobal S.A
Calea Bucuresti
Bl. 27B, Sc. 1, Ap. 10
Craiova, România
40 251-413496
www.museglobal.ro

EduLib, S.R.L.
Calea Bucuresti
Bl. 27B, Sc. 1, Ap. 2
Craiova, România
40 351-420970
www.edulib.com

Version: 1.0
Date: 28th September
2016
Author: EduLib, S.R.L.

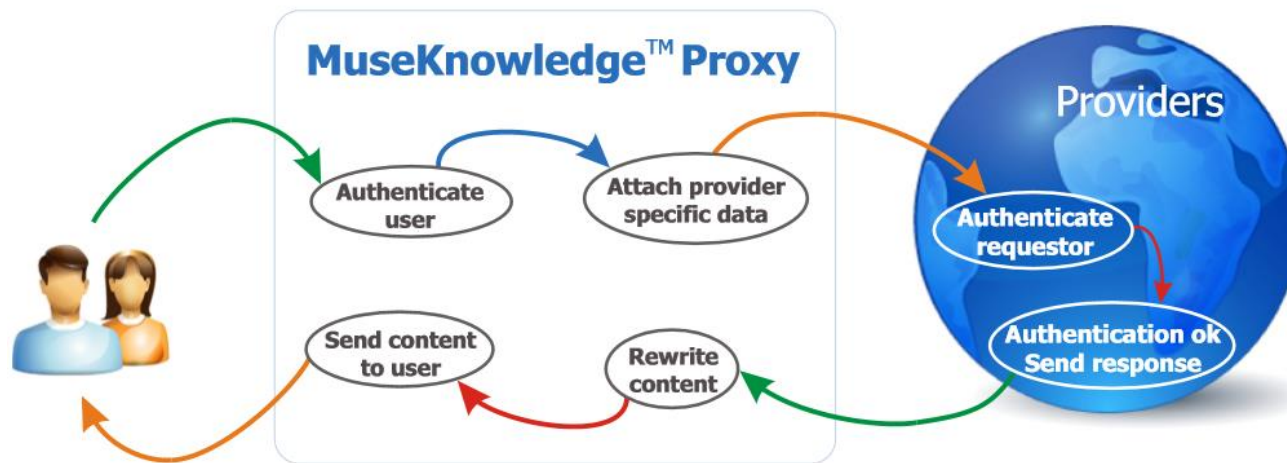
MuseKnowledge™ Proxy

Highly customizable multi-platform proxy server

- Easy to use and configure via the MuseKnowledge™ Proxy Administrator Console;
- Gateway to authenticated restricted content;
- Rewriting web server;
- Web Access Management (WAM);
- Proxy server and reverse proxy.

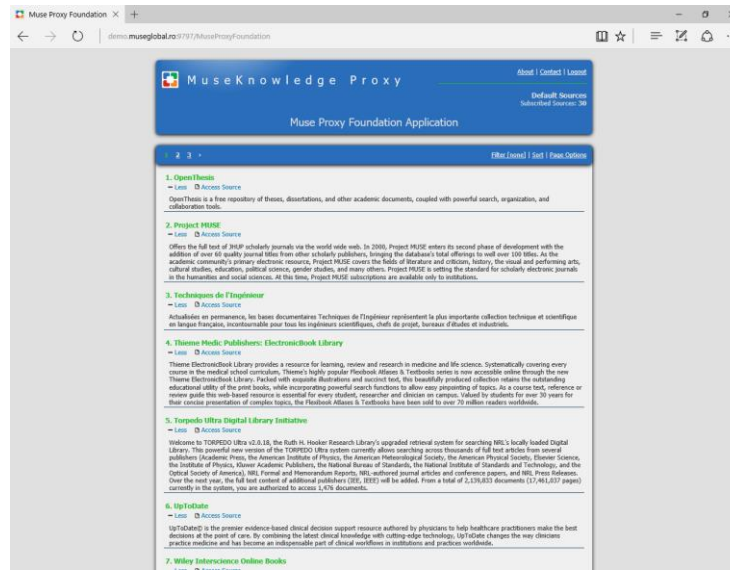


More than 10 years within the Muse Federated Search Platform to manage the authentication to resources and the navigation to full text



MuseKnowledge™ Proxy Access Methods

- By accessing the MuseKnowledge™ Proxy Application interface;



- Via entry point links:

`http(s)://${proxyHost}:${proxyPort}/ApplicationID?[groupID=${groupIDValue}&][${applicationAuthenticationParameters}&]url=${nonEncodedRemoteURL}`

- Via widget and form integration:

`http(s)://${proxyHost}:${proxyPort}/ApplicationID?groupID=${groupIDValue}[&${applicationAuthenticationParameters}]&action=source&sourceID=${sourceIDValue}[&nativeParams=GET|POST]&qurl=${encodedURL}`

MUSE



MuseKnowledge™ Proxy Authentication Types for Integration

- Direct Authentication – happens directly at MuseKnowledge™ Proxy
 - Username/Password, IP, LDAP, FTP, IMAP, SQL;
 - Referer – used in the Egyptian Knowledge Bank implementation as authentication method for the Muse Proxy Applications;
 - HMAC;
 - External HTTP;

Username/Password, IP, LDAP, FTP, IMAP, SQL and Referer authentications are presented in “MuseKnowledge Proxy Applications.pptx”.

- Indirect Authentication – happens on the identity provider side
 - SAML (detailed presentation “MuseKnowledge Proxy and SAML Authentication.pptx”): ADFS – Okta – Shibboleth – OpenAM - Efecte EIM or Ping Federate;
 - OAuth (detailed presentation “MuseKnowledge Proxy and OAuth Authentication.pptx”): BitBucket - CAS server using OAuth protocol - DropBox - Facebook - Foursquare - Github - Google - LinkedIn - Odnoklassniki - ORCiD - Paypal - Strava - Twitter - Vk - Windows Live - Word Press – Yahoo.



HMAC Authentication

- Stands for Keyed-Hash Message Authentication Code;
- Suitable for portal integrations to secure transparent log-on so that the end-user is not requested an explicit authentication to MuseKnowledge™ Proxy;
- Used as authentication method in the Egyptian Knowledge Bank implementation in the Muse Search application for the Researchers category;
- The HMAC links must be generated on server side;
- Secure, with multiple configurations options and limited lifespan, e.g. cannot be hijacked and executed after a certain time;
- Make a stronger authentication by enabling the use of the end-user IP, and/or referrer URL and/or browser user agent;
- Integrating server and MuseKnowledge™ Proxy server must have the time synchronized;
- The HMAC links must be generated only at the moment the end-user wants to access the MuseKnowledge™ Proxy link;
- Configuration file:
`${MUSE_HOME}/proxy/webcontexts/Applications/${AppID}/profiles/login/ProxyLoginModuleHMAC.xml`

More



HMAC Authentication

- Sample configuration of the HMAC authentication module

```
<HMAC_CONFIGURATION>
  <HMAC_PARAM>sig</HMAC_PARAM>
  <TS_PARAM>ts</TS_PARAM>
  <TS_EXPIRY>30</TS_EXPIRY>
  <SECRET>quiet</SECRET>
  <ALGORITHM>HmacSHA1</ALGORITHM>
  <SEPARATOR>.</SEPARATOR>
  <USED_PARAMS>
    <PARAM>userName</PARAM>
    <PARAM>ts</PARAM>
  </USED_PARAMS>
</HMAC_CONFIGURATION>
```

Name of the parameter to take the signature from

Validity of the time stamp (s)

Name of the parameter to take the time stamp from

The secret

The hashing algorithm: HmacSHA1, HmacSHA512, HmacMD5

Separator between parameters

List of parameters to be hashed, any combination of: username, ts, userAddress, userAgent, referer

MUSE



More

HMAC Authentication

- Signature configuration examples:
 - With `userName` and timestamp, e.g. `userName.ts`

```
<SEPARATOR>.</SEPARATOR>  
<USED_PARAMS>  
    <PARAM>userName</PARAM>  
    <PARAM>ts</PARAM>  
</USED_PARAMS>
```

The signature is the hashing of `userName.ts` string with the configured algorithm and secret.

Example: `userName` is `MuseProxyFoundation`, time stamp is `1470142967`, algorithm `HmacSHA1`, secret `quiet`, separator `.` .

String to sign: `MuseProxyFoundation.1470142967`

Resulted signature: `886db1ad254d1caf8823e6bf7298d209d9dba448`

HMAC link generated:

`http(s)://${proxyHost}:${proxyPort}/MuseProxyFoundation?userName=MuseProxyFoundation&ts=1470142967&sig=886db1ad254d1caf8823e6bf7298d209d9dba448`

MUSE



More

HMAC Authentication

- With `userName`, `userAddress` and `timestamp`, e.g. `userName.userAddress.ts`

```
<SEPARATOR>.</SEPARATOR>
<USED_PARAMS>
  <PARAM>userName</PARAM>
  <PARAM>userAddress</PARAM>
  <PARAM>ts</PARAM>
</USED_PARAMS>
```

The signature is the hashing of `userName.userAddress.ts` string with the configured algorithm and secret.

Example: `userName` is `MuseProxyFoundation`, time stamp is `1470142967`, algorithm `HmacSHA1`, IP address: `217.156.14.134`, secret `quiet`, separator `.`

String to sign: `MuseProxyFoundation.217.156.14.134.1470142967`

Resulted signature: `3bf4a9accf11b354adf9ec0b00abd8b95b86a422`

HMAC link generated:

`http(s)://${proxyHost}:${proxyPort}/MuseProxyFoundation?userName=MuseProxyFoundation&ts=1470142967&sig=3bf4a9accf11b354adf9ec0b00abd8b95b86a422`

MUSE



More

HMAC Authentication

- Workflow

- The end-user clicks on a link in the portal;
- Dynamically the HMAC link is built:

http(s)://\${proxyHost}:\${proxyPort}/MuseProxyFoundation?userName=MuseProxyFoundation&ts=1470142967&sig=3bf4a9accf11b354adf9ec0b00abd8b95b86a422

- The request reaches MuseKnowledge™ Proxy which begins the authentication process:
 - Based on the HMAC authentication module configuration file it constructs the signature and compares it with the one received in the request;
 - If they are different the authentication fails;
 - Otherwise, if they are equal and the difference between the current timestamp and the one received as parameter in the request is smaller than the configured expiry value (TS_EXPIRY) then the authentication succeeds;
 - Otherwise if the time difference is bigger than the configured expiry time, then the authentication fails.

MUSE



More

HMAC Authentication

- Configure the HMAC Authentication

- Create the new application as copy of the MuseProxyFoundation template - MuseProxyFoundationHMAC;

- Edit:

`${MUSE_HOME}\proxy\webcontexts\Applications\MuseProxyFoundationHMAC\profiles\AuthenticationGroups.xml`

- Remove all content of node `/ICE-CONFIG/AUTHENTICATION_GROUPS/AUTHENTICATION_GROUP/AUTHENTICATIONS;`

- Add in the same node:

```
<AUTHENTICATION>
  <IDENTIFIER>9</IDENTIFIER>
  <LEVEL>requisite</LEVEL>
  <CLASS>com.edulib.muse.proxy.authentication.modules.ProxyLoginModuleHMAC</CLASS>
  <HANDLER>
    <CLASS>com.edulib.muse.proxy.authentication.modules.ProxyLoginModuleHMACDataHandlerXml</CLASS>
    <PARAMETERS>
      <CONFIGURATION_FILE>${WEB_CONTEXT_HOME}/profiles/login/ProxyLoginModuleHMAC.xml</CONFIGURATION_FILE>
    </PARAMETERS>
  </HANDLER>
</AUTHENTICATION>
```

- Refresh the applications properties via the *Muse Proxy Administrator Console* -> *Advanced* left menu section -> *Operations* item -> *Refresh Applications* button.

MUSE



More

HMAC Authentication

- Configure the HMAC Authentication
 - Detailed configuration steps also available as [FAQ](#);
 - High flexibility in combining with other authentication methods available. For example:
 - HMAC or UserName/Password authentication;
 - HMAC or IP authentication;
 - etc;
 - Several implementation snippets are available for various technologies:
 - PHP – Consult FAQ entry [here](#) for more details;
 - Java Servlet – Consult FAQ entry [here](#) for more details;
 - .Net (ASPX) – Consult FAQ entry [here](#) for more details;

MUSE



More

HMAC Authentication

- HMAC Link Generator tool – use it for testing the HMAC authentication setup;
 - Available in the MuseKnowledge™ Proxy Administration Console, *Utilities* -> *HMAC Link Generator*;
 - Configurable rewriting proxy (Proxy URL);
 - Possibility to form an entry point link by providing the URL to be proxied;
 - Select the Muse Proxy Application to be used;
 - Add/Remove signature parameters;
 - Change the order of the signature parameters by simply dragging them;
 - Configure the parameters separator (default is .);
 - Configure the value for the secret;
 - Choose the encryption algorithm from the available ones;
 - Generate the link on demand and test it by click of a button.

The screenshot shows the 'HMAC Link Generator' utility in the MuseProxy Administration Console. The interface includes the following fields and controls:

- Proxy URL:**
- URL to be proxied:**
- Application:**
- Parameters/headers to use for HMAC:**
 - userName:
 - ts:
 - userAgent:
 - referer:
 - userAddress:
- New Parameter:**
- Separator:**
- Secret:**
- Algorithm:**
- Message to sign:**
-
- Generated Hmac code:**
- Generated Hmac link:**
-

MUSE



More

HTTP External Authentication

- Performs authentication against an external HTTP source such as an existent portal login point via standard CGI, REST service or standard HTTP Basic access authentication;
- No need for database or integration development work;
- Validation consists in detecting a sequence that corresponds to successful or unsuccessful logon;
- No need to modify the external service response to include specific authentication elements;
- Configuration File:
``${MUSE_HOME}/proxy/webcontexts/Applications/AppID/profiles/login/ProxyLoginModuleExtHTTP.xml`
- Parameters: configurable, and the out of the box configuration is using `userName` and `userPassword`;
- Regular expression extractors for identifying the successful or unsuccessful logon.

MUSE



More

HTTP External Authentication

End-user

End-user enters his personal authentication details

Login

Type in your credentials to access the subscribed sources.

User Name:

Password:

Muse Proxy Applications provide a fully configurable interface allowing administrators to setup remote or local Data Sources ("Sources") and where end users enjoy a single sign-on for all of the subscribed data.

Library Administrators can add Sources to the Muse Proxy Application depending of licensing rights. Each Source stores the necessary code and credentials to get to the target Data Source. In this way, MuseKnowledge Proxy enables end users to reach all configured targets with a single sign-on and conceals target authentication credentials.

MuseKnowledge™ Proxy uses the details to perform authentication to the configured customer portal

The Portal UNIVERSITY OF LEEDS

Username:

Password:

» [Get your username and password](#)
» [Forgot your password?](#)

Authorised users only.
Users must comply with University Use of Computer Systems Policy

» [Problems using the Portal](#)
» [Contact IT Service Desk](#)
» [IT services status](#)

Journeys & Travels
Skagofoss Iceland, courtesy of Nadia Hartanto

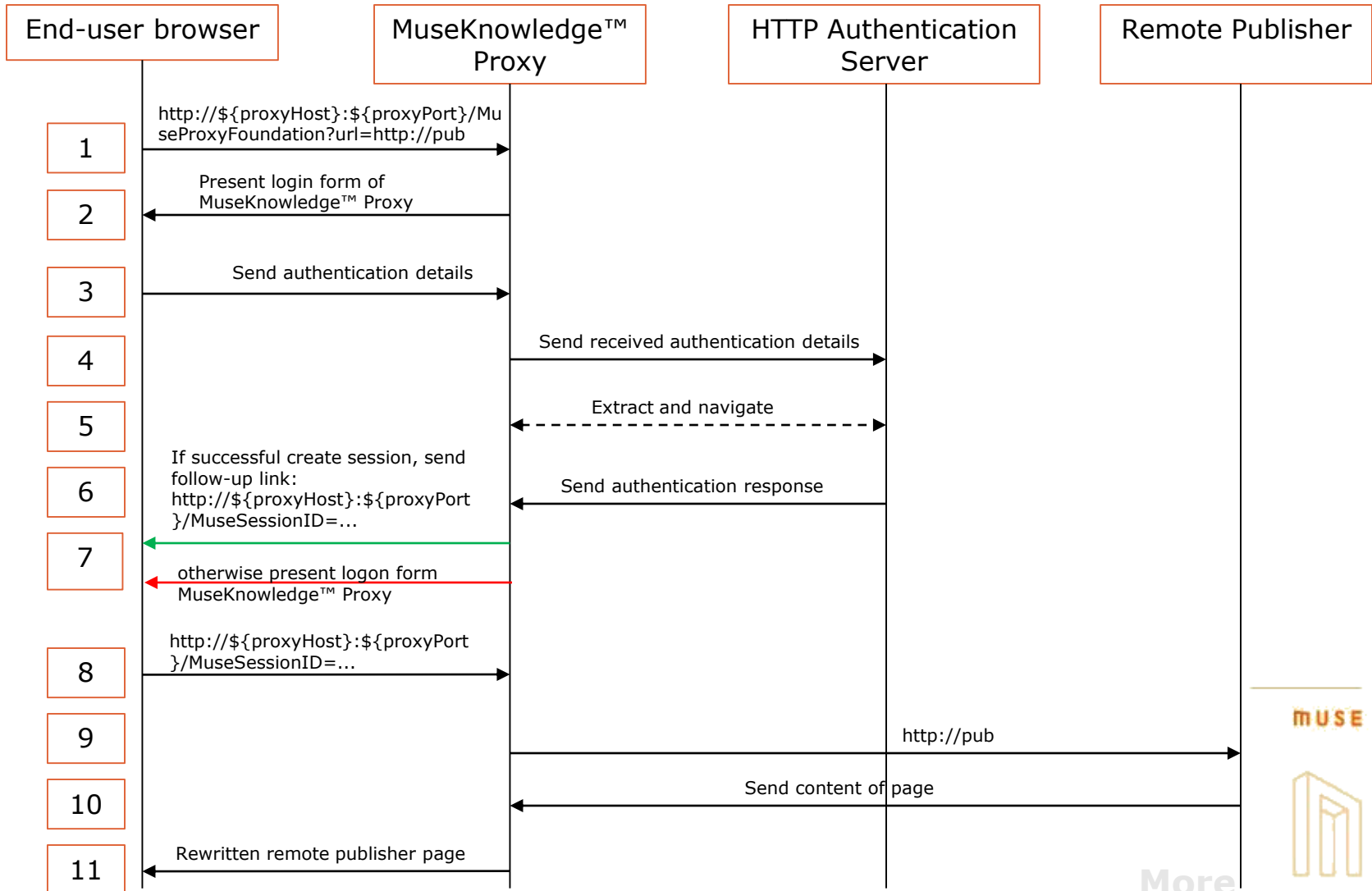
Copyright and Legal | Accessibility | Privacy & cookies
Copyright © 2016 University of Leeds

MUSE



More

HTTP External Authentication



HTTP External Authentication

- Example configuration using authentication against Muse Search via its XML API from demo.museglobal.ro;

```
<ExtHTTP_CONFIGURATION>
  <URL>http://demo.museglobal.ro:8000/muse/servlet/MusePeer</URL>
  <POST_PARAMETERS><![CDATA[action=logon&userID=${userName}&userPwd=${userPassword}&useProperties=false&
    templateFile=xml%2Findex.xml&errorTemplate=xml%2Ferror.xml]]></POST_PARAMETERS>
  <EXTRACTOR ref="success"><![CDATA[<SESSION_ID>([^\<]+)</SESSION_ID>]]></EXTRACTOR>
  <SSL_TRUST_ALL>true</SSL_TRUST_ALL>
  <FOLLOW_REDIRECTS>true</FOLLOW_REDIRECTS>
  <USED_PARAMS process="urlEncode">
    <PARAM>userName</PARAM>
    <PARAM>userPassword</PARAM>
  </USED_PARAMS>
  <LOG_USER_ID>${userName}</LOG_USER_ID>
</ExtHTTP_CONFIGURATION>
```

MUSE



References

- Muse Proxy.pdf;
- Muse Proxy Advanced Configuration.pdf;
- Muse Proxy Administrator Console.pdf;
- FAQ “[How to set up HMAC authentication on a Muse Proxy application?](#)”
- FAQ “[How can I generate HMAC Muse Proxy proxified links in PHP?](#)”
- FAQ “[How can I generate HMAC Muse Proxy proxified links in Java?](#)”
- FAQ “[How can I generate HMAC Muse Proxy proxified links in .Net?](#)”

MUSE





MUSE

MuseKnowledge™ Proxy Integration