



Muse LinkResolver

29 June 2023

Document Version 0.3
Muse 2.8.0.1



Notice

No part of this publication may be reproduced stored in a retrieval system, or transmitted, in any form or by any means, without the prior written permission of MuseGlobal Inc.

Disclaimer

MUSEGLOBAL, INC. MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OR MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE.

Trademarks

MUSE IS A REGISTERED TRADEMARK OF MUSEGLOBAL, INC. OTHER PRODUCT NAMES AND SERVICE NAMES ARE THE TRADEMARKS OR REGISTERED TRADEMARKS OF THEIR RESPECTIVE OWNERS AND ARE USED FOR IDENTIFICATION ONLY.

www.museglobal.com







Table of Contents

1.0 Overview	5
1.1 About OpenURL	5
1.2 About Muse LinkResolver	6
1.2.1 Scenario exemple	6
2.0 Muse LinkResolver implementation	9
2.1 Requests Workflow	9
2.2 Authentication	10
2.3 Titles A to Z	11
2.4 Interface customization and internationalization	12
2.5 Accessing Muse LinkResolver links	13
2.6 Muse LinkResolver application configuration files	13
2.7 Muse LinkResolver Administration Interface	14
2.7.1 LinkResolver General Configuration	15
2.7.2 Authentication	16
3.0 OpenURL clients integration	19
3.1 MuseKnowledge Search application integration with Muse LinkResolver	19
3.1.1 Steps performed for the MuseKnowledge Search application	19
3.1.2 Steps performed for the Muse LinkResolver application	20
1.0 OpenURL versions	21
1.1 OpenURL v0.1	21





1.0

Overview

Link resolvers provide customized lists with title-level metadata to libraries for ease of content discoverability and full-text linking. When an institution subscribes to multiple databases and full-text resources, it can be a challenge to determine if the full-text of an article cited in one database is available in another resource. Link resolving software makes this task easier by acting as a bridge between databases; end-users easily go from a journal citation in one database to the full-text of the journal in another database.

Muse LinkResolver is a software application that uses the OpenURL protocol to provide seamless access to electronic full-text content across all databases and vendors for a provided citation. It uses the publisher knowledge base from Muse Central Index for finding the appropriate linking options.

1.1 About OpenURL

The OpenURL is a protocol for interoperability between an information resource and a service component that offers localized services in an open linking environment. It is in effect an actionable URL that transports metadata or keys to access metadata for the object for which the OpenURL is provided. The target of the OpenURL is the user's institutional service component (ISC). The remainder of the OpenURL transports the object's metadata. The format and more specification for OpenURL can be found at http://www.niso.org/committees/committee_ax.html.

The major application of the OpenURL Framework is to enable context-sensitive linking from a reference in a scholarly information system to resources relevant to the referenced item. Prior to the publication of the OpenURL Framework Standard, applications have been based on the draft OpenURL (now known as version 0.1).

The traditional use of the OpenURL Framework is when a user clicks a link or button in an HTML page, typically within an 'abstracting and indexing' database application or the reference list of an electronic journal article. The result of activating this link is to transport a description of a scholarly resource, such as a journal article, to a linking server along with information about the dynamic context



of the reference. Passing the dynamic context enables the provision of a list of relevant services appropriate for the user. The Transport is based on HTTP(S) GET or POST and is referred to as "an OpenURL". Using OpenURL version 0.1, the information, or "payload" of the OpenURL, is transported inline as the "query string" of a URL. The Key/Encoded-Value Format provides a similar means of transporting a description of a reference and its context.

In 2005, a revised version of OpenURL (version 1.0) became ANSI/NISO standard Z39.88-2004. The new standard provided a framework for describing new formats, as well as defining XML versions of the various formats.

Revisions:

- ✎ NISO OpenURL version 0.1 (2000-05-16).
- ✎ NISO OpenURL version 1.0f (2003-03-18)
- ✎ ANSI/NISO Z39.88-2004.
- ✎ ANSI/NISO Z39.88-2004 [R2010]

1.2 About Muse LinkResolver

Muse LinkResolver implements the OpenURL standard and currently supports version 0.1. Version 1.0 will be supported in future versions.

The application can be used independently, serving applications that generate OpenURL requests, or part of the Muse Platform, integrated with the Muse Search Application interface.

From technical point of view, the Muse LinkResolver is a web service which can be accessed through a HTTP request and which returns relevant information about an article, bookitem, journal, book, etc., information stored in Muse Central Index.

The requester must provide in the HTTP request a list of parameters using the OpenURL standard syntax v0.1. The Muse LinkResolver application validates these parameters and performs a search in Muse Central Index to find relevant information and responds with a HTML page where the information is displayed.

1.2.1 Scenario example

The scenario described below refers to the integration of the Muse LinkResolver with the Muse Search Application. This means that the application responds to an OpenURL request initiated from the Muse



Search Application, from the search results. A typical use case is when the end-user is looking for the full text of an article revealed from searching an abstract/citation database.

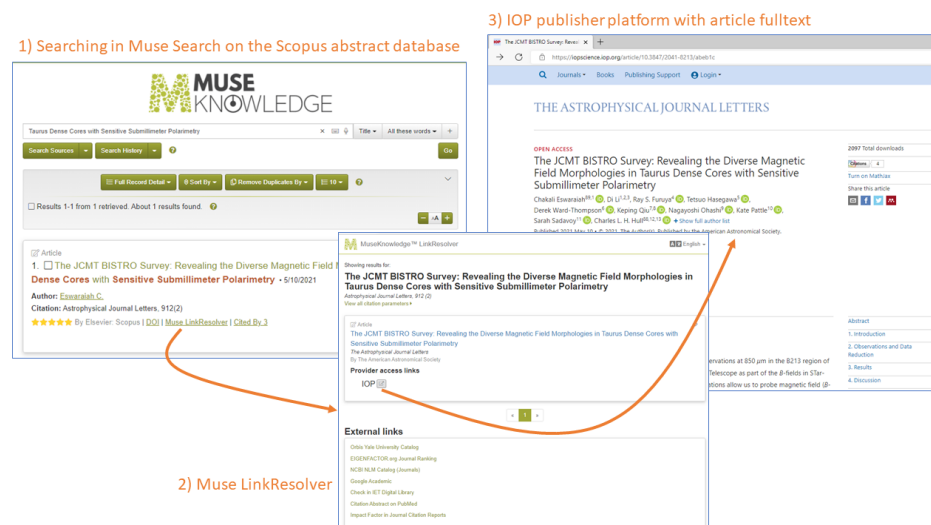
- ✎ The user performs a search in the Muse Search Application and obtains a list of results from a citation database. He notices a result of interest and wants to read the fulltext article. A link to fulltext is not available in the metadata elements of that result.
- ✎ The user clicks the Muse LinkResolver link (which is an OpenURL containing citation elements from the result's metadata) provided in the result's display.

As a result of this action, in a new tab the Muse LinkResolver application opens. The page contains a result or a list of results with linking options to various publisher platforms where the article fulltext can be found. The metadata displayed in the results may be as follows:

- ✎ **title / journal / book / bookitem / conference;**
- ✎ **isbn / issn / eissn / eisbn , volume, issue, etc.;**
- ✎ a list of URLs (URL, Persistent URL, etc.) of (**article / book / bookitem / journal,** etc.);
- ✎ or a DOI;
- ✎ other metadata.

if this information exists in linkresolver's knowledgebase.

This exploratory workflow is depicted in the below image:





2.0

Muse LinkResolver implementation

The Muse LinkResolver application is a web application running in a web container server such as the Apache Tomcat server. It is built using the SpringBoot framework and the interface uses the Thymeleaf templating engine with Spring. The interface is full responsive on all mobile devices, using technologies such as Bootstrap and jQuery.

2.1 Requests Workflow

Muse LinkResolver is mainly used with discovery applications, to provide linking options to help users reach full-text content. The below result example is taken from the Muse Search application.

Clicking the **Muse LinkResolver** link from search results opens the Muse LinkResolver application in a new browser tab.



Figure 1. Request Flow



Muse LinkResolver performs a search for article metadata in the Muse Central Index using the values received in the OpenURL link. If an article result is found, it is displayed in the interface. If no article is found matching the provided query, another search is done for titles metadata. If titles results are found, they are displayed as alternatives for the received citation details.

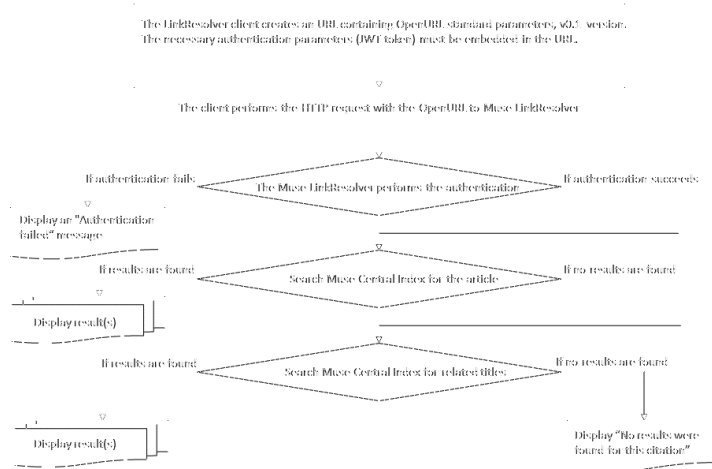


Figure 2. Flowchart

2.2 Authentication

Currently the Muse LinkResolver requires authentication to serve the OpenURL requests. If authentication details are not provided in the request, an error page is displayed.

The application is intended to support multiple configurable authentication methods (e.g. username/password, IP, SSO, etc.), but currently, only token based authentication (JSON Web Token JWT - <https://jwt.io/>) and IP authentication methods are available. The following configuration methods are available application's configuration file `authentication.properties` :

- 1 **NONE**. No authentication required, meaning anonymous access.

`authentication.type=NONE`

- 2 **JWT**. JWTs are used as a secure way to authenticate the end-users against Muse LinkResolver.



Typically, a secret, is used by the issuer (the search application from where the Muse LinkResolver URL is accessed) to sign the JWT. The Muse LinkResolver application receiving the JWT verifies the signature and if successful, it fulfills the request. If fails, an authentication failure message is issued. The configuration properties for enabling and configuring JWT authentication are below:

```
authentication.type=JWT
#The jwt secret key for authentication
authentication.jwt.secretKey=mySecretKey
#The jwt token for authentication
authentication.jwt.tokenParam=token
```

- 3 IP. IP authentication, to authenticate incoming requests against a specified list of allowed IPs/subnets or domain patterns. The configuration properties for enabling and configuring IP authentication are below:

```
authentication.type=IP
#Specify a list of IPs / IP patterns, hostnames or hostname patterns
which are allowed
authentication.ip.list-allow=*.museglobal.ro,host.docker.internal,127.0.0.1,0:0:0:0:0:0:1
#Specify a list of IPs / IP patterns, hostnames or hostname patterns
which are denied
authentication.ip.list-deny=
```

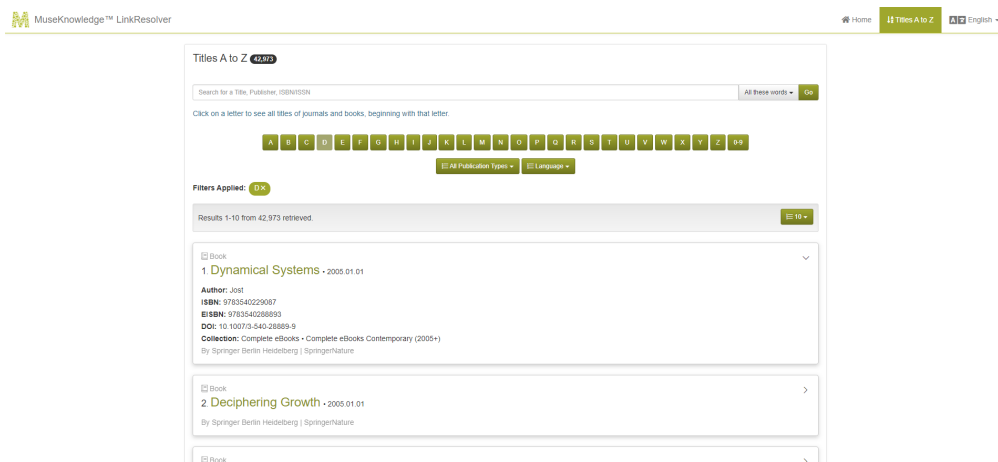
- 4 Administrator user details. This section contains the authentication details for the administration console: the username, password and the IP based ACL rules.

```
administrator.username=admin
administrator.password=HS&#hadi74cwq
administrator.ip-list-allow=*.museglobal.ro,host.docker.internal,127.0.0.1,0:0:0:0:0:0:1
administrator.ip-list-deny=
```

2.3 Titles A to Z



A Titles A to Z functionality is available, accessible from the application's main menu. It provides support for browsing the available titles alphabetically or by filtering by the Publication Type or Language. A keyword search can also be used for locating the titles of interest more quickly.



The Titles A to Z page can be disabled from configuration:

```
display.showAtozLink=false
```

in the `linkresolver-application.properties` file.

2.4 Interface customization and internationalization

The interface of the Muse LinkResolver application can be customized through CSS files.

A set of predefined themes is available: Muse Default Theme (ID `mgb`), Bootstrap Default Theme (ID `bootstrap`), Dark (ID `dark`), White (ID `white`), Dark Blue (ID `darkblue`), Purple (ID `purple`), Gray (ID `gray`), Brown (ID `brown`). The Muse LinkResolver application will load automatically the theme with identifier as received in the `&aspect` parameter. Below is a sample OpenURL link for the MuseLinkResolver which will display theme Purple:

```
https://link.domain.com/linkresolver/openurl/v0.1?title=Lomonosov+and+World+Science&isbn=9789027710628&genre=book&aufirst=Kapitza&date=1980-01-01&publisher=Springer&sid=MuseKnowledgeFoundation&aspect=purple
```

The Muse Search application is sending automatically the code of the theme in use to ensure the same



theme is loaded in Muse LinkResolver for a smooth interface experience.

The application supports internationalization, a language swithing functionality is available. Currently the following languages are available: English (default), Romanian, Arabic, Greek, Spanish, French, Chinese and Dutch. The interface language can be selected using the following request parameter **&lang**.

2.5 Accessing Muse LinkResolver links

To access the publisher links using institutional subscription, a rewriting proxy such as Muse Proxy must be used to prefix all URLs provided by the LinkResolver. The proxy prefix URL is specified in the application's configuration file - **linkresolver-application.properties**. Below is an example of configuration with Muse Proxy:

```
url.rewritting.active=true
url.rewritting.proxy-prefix=https://demopr.x.museglobal.ro/Demo
url.rewritting.field.identifier-url=${url.rewritting.proxy-prefix}?qurl=#{encodeURLParam(${identifier-url})}
url.rewritting.field.identifier-persistent-url=${url.rewritting.proxy-prefix}?qurl=#{encodeURLParam(${identifier-persistent-url})}
url.rewritting.field.identifier-doi-url=${url.rewritting.proxy-prefix}?qurl=#{encodeURLParam(${identifier-doi-url})}
url.rewritting.field.identifier-abstract-url=${url.rewritting.proxy-prefix}?qurl=#{encodeURLParam(${identifier-abstract-url})}
url.rewritting.field.identifier-pdf-url=${url.rewritting.proxy-prefix}?qurl=#{encodeURLParam(${identifier-pdf-url})}
```

When clicking a proxified URL the flow is passed to the rewriting proxy which provides seamless access into the publisher's platform.

2.6 Muse LinkResolver application configuration files

The Muse LinkResolver application has several configuration files. Alternatively one can use a config server (GIT or database) to store the configuration files.

The main configuration file is **application.properties** and contains the following configuration properties:



- ✎ the application name;
- ✎ the external configuration files if these files are used from the local filesystem;
- ✎ the address of the configuration server and the name of the external configuration files stored on a GIT server or on a database;
- ✎ the path to the `i18n` files (internationalization);
- ✎ other internal properties.

The `linkresolver-application.properties` file is one of the external properties file which can be loaded from local machine or from a GIT server in case a config server is used. This property file contains the following:

- ✎ the URL for the index service, part of Muse Central Index;
- ✎ the names of the indices storing the articles and titles metadata;
- ✎ the URI for the database, part of Muse Central Index;
- ✎ the application root path;
- ✎ the rewriting URL parameters;
- ✎ the security parameters;
- ✎ the `perPage` and `maxResults` parameters for the search on the Muse Central Index.

The `linkresolver-mapping.properties` file is one of the external properties file which can be loaded from the local filesystem or from a GIT server in case a config server is used. This property file contains the following:

- ✎ a map containing `openURL` parameter from request and their mapping for the Solr field;
- ✎ a map used to validate a requested object type (bookitem, book, preprint, article, journal, conference, proceeding);
- ✎ a map used to associate the requested object type and the valuelist of types from Muse Central Index;
- ✎ a map used to display the result metadata.

2.7 Muse LinkResolver Administration Interface

The Muse LinkResolver application has an administration interface accessible at the following URL:

`https://link.domain.com/linkresolver/openurl/login`



The authentication credentials (username and password) are stored in the **authentication.properties** configuration file. The authentication is multi-factor, besides the username/password, the user's IP address is also considered. In the **authentication.properties** configuration file can be specified lists with IP(s)/ranges/patterns/domains to be allowed or denied.

The interface of the Muse LinkResolver Administration console has a side main menu with 3 sections: LinkResolver General Configuration, AtoZ and Authentication.

2.7.1 LinkResolver General Configuration

In this section one can configure general aspects of the application, such as the availability of features: the A to Z and the External Links. And a read-only view of the connectivity details for Muse Central Index (the Solr index and MongoDB database).

2.7.1.1 Rewrite URLs

The linkresolver results URLs rewriting behavior can be configured in this section. The rewriting can be enabled/disabled and rules can be specified for each type of URL:

Rewrite URLs

Here you can enable or disable the proxy rewriting mechanism, to specify a rewriting proxy prefix and patterns for articles/titles result URLs. These settings apply for the Titles A to Z functionality too.

URL Rewriting Active: ?

☒ True ☐ False

Proxy Prefix: ?

Abstract URL: ?

Pdf URL: ?

DOI URL: ?

Persistent URL: ?

URL: ?

2.7.1.2 Display Fields

This section allows the mapping of record's fields into keys to be displayed in the application's interface.



Rewrite URLs

Here you can enable or disable the proxy rewriting mechanism, to specify a rewriting proxy prefix and patterns for articles/titles result URLs. These settings apply for the Titles A to Z functionality too.

URL Rewriting Active: ?

☒ True ☐ False

Proxy Prefix: ?

Abstract URL: ?

Pdf URL: ?

DOI URL: ?

Persistent URL: ?

URL: ?

2.7.1.3 External Links

The management of the external links that appear into the Muse LinkResolver is done in this section. New external links configurations can be added and existing ones can be edited/deleted.

External Links

The below table contain a list of external service providers. To add a new external service provider you have to press the "Add External Link" button and to fill the popup fields.

Add External Link									
Show 10 entries		Search:							
#	Name	Rewriting Active	Rewriting Proxy Prefix	Rewriting Pattern	Search url	Description	Http Method		
1	yale-university-library	false	https://demopr.museglobal.ro/MuseKnowledgeFoundation	https://demopr.museglobal.ro/MuseKnowledgeFoundation?qurl=#{encodeURIComponent(\$identifier-abstract-uri)}	https://orbis.library.yale.edu/vwebv/search	Orbis Yale University Catalog	GET		
2	eigen-factor-journal-ranking	false	https://demopr.museglobal.ro/MuseKnowledgeFoundation	https://demopr.museglobal.ro/MuseKnowledgeFoundation?qurl=#{encodeURIComponent(\$identifier-abstract-uri)}	http://www.eigenfactor.org/rankings.php/rankings.php	EIGENFACTOR.org Journal Ranking	GET		
3	nim-catalog	false	https://demopr.museglobal.ro/MuseKnowledgeFoundation	https://demopr.museglobal.ro/MuseKnowledgeFoundation?qurl=#{encodeURIComponent(\$identifier-abstract-uri)}	https://www.ncbi.nlm.nih.gov/ncat/catalog/	NCBI NLM Catalog (Journals)	GET		
4	ncbi-bookshelf	false	https://demopr.museglobal.ro/MuseKnowledgeFoundation	https://demopr.museglobal.ro/MuseKnowledgeFoundation?qurl=#{encodeURIComponent(\$identifier-abstract-uri)}	https://www.ncbi.nlm.nih.gov/books	NCBI Bookshelf	GET		
5	google-books	false	https://demopr.museglobal.ro/MuseKnowledgeFoundation	https://demopr.museglobal.ro/MuseKnowledgeFoundation?qurl=#{encodeURIComponent(\$identifier-abstract-uri)}	https://books.google.com/books	Google Books	GET		
6	google-academic	false	https://demopr.museglobal.ro/MuseKnowledgeFoundation	https://demopr.museglobal.ro/MuseKnowledgeFoundation?qurl=#{encodeURIComponent(\$identifier-abstract-uri)}	https://scholar.google.com/scholar	Google Academic	GET		
7	iet-digital-library	false	https://demopr.museglobal.ro/MuseKnowledgeFoundation	https://demopr.museglobal.ro/MuseKnowledgeFoundation?qurl=#{encodeURIComponent(\$identifier-abstract-uri)}	https://digital-library.ietel.org/search	Check in IET Digital Library	GET		
8	pubmed	true	https://demopr.museglobal.ro/MuseKnowledgeFoundation	https://demopr.museglobal.ro/MuseKnowledgeFoundation?qurl=#{encodeURIComponent(\$identifier-abstract-uri)}	https://pubmed.ncbi.nlm.nih.gov/	Citation Abstract on PubMed	GET		
9	clarivate	true	https://demopr.museglobal.ro/MuseKnowledgeFoundation	https://demopr.museglobal.ro/MuseKnowledgeFoundation?qurl=#{encodeURIComponent(\$identifier-abstract-uri)}	http://gateway.webofknowledge.com/gateway/Gateway.cgi	Impact Factor in Journal Citation Reports	GET		

Showing 1 to 9 of 9 entries

Previous **1** Next

2.7.2 Authentication



The authentication method for the Muse LinkResolver application can be configured in this location. Currently the following authentication methods are available: by JWT and by IP. No authentication can be configured as well.

Type

☐ NONE ☒ JWT ☐ IP

JWT Secret Key: ?

dw213febe43vvc

JWT Token Param: ?

token

Type

☐ NONE ☐ JWT ☒ IP

IP List Allow: ?

*.museglobal.ro,host.docker.internal,127.0.0.1,0:0:0:0:0:0:1

IP List Deny: ?

The access details for the administrator user can be edited in this section.





3.0

OpenURL clients integration

This chapter describes how to integrate Muse LinkResolver with OpenURL clients and the necessary instructions to be made on both the client side and on the LinkResolver application side.

3.1 MuseKnowledge Search application integration with Muse LinkResolver

The MuseKnowledge Search application has complete support for integrating a LinkResolver application supporting the OpenURL standard v0.1 .

3.1.1 Steps performed for the MuseKnowledge Search application

The configuration of a LinkResolver for the MuseKnowledge Search application is done from the Muse Console for Applications Administration, more exactly from the **Application General Settings** -> **OpenURL Resolver** section.

The following details need to be configured:

- ✎ The URL for the Muse LinkResolver application, e.g.
`https://demo.museglobal.ro/linkresolver.`
- ✎ The HTTP method to be used when linking from the MuseKnowledge Search application interface into the Muse LinkResolver application. Both **GET** and **POST** methods are supported. It is recommended to use **POST** as it is not limited by the length of the payload.
- ✎ The authentication type. **JWT** is the only authentication method that requires configurations on the client side as well. Upon selection of **JWT**, the **Secret key** value must be provided. It must be the same value as configured in the Muse LinkResolver application.
- ✎ How the link for the Muse LinkResolver to be displayed in the results listing for each result. The



only options available are to display the link using an image/icon from a specified URL, or a simple text.

3.1.2 Steps performed for the Muse LinkResolver application

The only setting to be made in the Muse LinkResolver application for integration with the MuseKnowledge Search application is the value for the secret for the **JWT** token used in the authentication process. More exactly the secret value must be the same in both applications.

The value for the JWT secret is specified in the following Muse LinkResolver configuration file, **jwt.secretKey** key:

```
${MUSE_HOME}/linkresolver/WEB-INF/classes/linkresolver-application.properties
```



1.0

OpenURL versions

The next section provides detailed information about the OpenURL syntax for the version implemented by Muse LinkResolver.

1.1 OpenURL v0.1

The OpenURL syntax description uses an HTTP GET request format. However, the same syntax can also be used in an HTTP POST format.

OpenURL ::= BASE-URL '?' QUERY QUERY ::= DESCRIPTION ('&' DESCRIPTION)

- ✎ **BASE-URL** is the URL of a service-component that can take an OpenURL as input.
- ✎ **DESCRIPTION** describes the origin of the transported metadata-object as well as the metadata-object itself.
- ✎ If multiple objects are transported over the OpenURL, their **DESCRIPTION** must be delimited by two ampersands.

Example:

- ✎ A **BASE-URL** could be `http://sfxserver.uni.edu/sfxmenu`
- ✎ The **BASE-URL** will depend on the user (or its institution) and can -- for instance -- become known to the information provider via the CookiePusher mechanism.

DESCRIPTION ::= (ORIGIN-DESCRIPTION '&')? OBJECT-DESCRIPTION | OBJECT-DESCRIPTION ('&' ORIGIN-DESCRIPTION)?

- ✎ **OBJECT-DESCRIPTION** contains information about the metadata-object transported in the OpenURL.
- ✎ **ORIGIN-DESCRIPTION** contains information about the information system where the transported metadata-object originates. It describes the system that inserts the OpenURL.



The OpenURL must transport at least one object. As such the OpenURL must contain at least one **OBJECT-DESCRIPTION**.

- ✎ The order in which **OBJECT-DESCRIPTION** and **ORIGIN-DESCRIPTION** are provided is not significant.

ORIGIN-DESCRIPTION ::= sid '=' VendorID ':' DatabaseID

VendorID ::= (ALPHANUM)+

DatabaseID ::= (ALPHANUM | ESCAPED)+

- ✎ The **ORIGIN-DESCRIPTION** consists of the sid tag-name (service identifier) and a corresponding tag-value. This tag-value consists of two parts that are separated by a colon. The part before the colon is the identifier of the vendor of the information service where the metadata originates. The part of the tag-value following the colon is the identifier of the database within the vendor's information service where the metadata originates. The colon is provided 'as is', meaning in a non Escape encoded form.
- ✎ It is highly recommended to provide an **ORIGIN-DESCRIPTION**. If the **OBJECT-DESCRIPTION** contains a **LOCAL-IDENTIFIER-ZONE** then the provision of **ORIGIN-DESCRIPTION** is mandatory.

Examples of **ORIGIN-DESCRIPTION**

✎ **sid=Ovid:Medline**

✎ **sid=ERL:BX4**

✎ **sid=EBSCO:MFA**

OBJECT-DESCRIPTION ::= ZONE ('&' ZONE) *

ZONE ::= (GLOBAL-IDENTIFIER-ZONE | OBJECT-METADATA-ZONE | LOCAL-IDENTIFIER-ZONE)

GLOBAL-IDENTIFIER-ZONE ::= 'id' '=' GLOBAL-NAMESPACE ':' GLOBAL-IDENTIFIER ('&' 'id' '=' GLOBAL-NAMESPACE ':' GLOBAL-IDENTIFIER) *

GLOBAL-NAMESPACE ::= ('doi' | 'pmid' | 'bibcode' | 'oai')

GLOBAL-IDENTIFIER ::= VCHAR+

The **GLOBAL-IDENTIFIER-ZONE** contains identifiers of global namespaces and the corresponding identifiers of the transported object within these global namespaces. Identifiers that only have significance in local namespaces -- such as the identifier of a result in an institutional implementation of an A&I database -- do not fit into this zone. They belong in the **LOCAL-IDENTIFIER-ZONE**.



Example:

- ✎ A GLOBAL-IDENTIFIER-ZONE can be: `id=doi:123/345678&id=pmid:202123`
- ✎ A valid OpenURL -- before the mandatory Escape encoding -- is:
`http://sfxserver.uni.edu/sfxmenu?id=doi:123/345678&id=pmid:202123` This OpenURL transports two global identifiers that uniquely define the same metadata-object.
- ✎ A valid OpenURL -- before the mandatory Escape encoding -- for a preprint that resides in an archive that complies with the Santa Fe Convention of the Open Archives initiative is:
`http://sfxserver.uni.edu/sfxmenu?id=oai:arXiv:physics/0003005`

OBJECT-METADATA-ZONE ::= META-TAG '=' META-VALUE (& META-TAG '=' META-VALUE) *

META-TAG ::= ('genre' | 'aulast' | 'aufirst' | 'auinit' | 'auinit1' | 'auinitm' | 'coden' | 'issn' | 'eissn' | 'isbn' | 'title' | 'stitle' | 'atitle' | 'volume' | 'part' | 'issue' | 'spage' | 'epage' | 'pages' | 'artnum' | 'sici' | 'bici' | 'ssn' | 'quarter' | 'date')

META-VALUE ::= VCHAR+

Example:

- ✎ An OBJECT-METADATA-ZONE can be:
`issn=1234-5678&date=1998&volume=12&issue=2&spage=134`
- ✎ A valid OpenURL can be :
`http://sfxserver.uni.edu/sfxmenu?issn=1234-5678&date=1998&volume=12&issue=2&spage=134.`

LOCAL-IDENTIFIER-ZONE ::= 'pid' '=' VCHAR+

The **LOCAL-IDENTIFIER-ZONE** is introduced in order to allow for the transportation of metadata in formats that are specific to the originating information system, and that can not be expressed in the standardized syntax proposed for the **OBJECT-METADATA-ZONE**.

Example:

- ✎ A LOCAL-IDENTIFIER-ZONE can be: `pid=<author>Smith, Paul ; Klein, Calvin</author>&<yr>98</yr>`
- ✎ An OpenURL containing the above **LOCAL-IDENTIFIER-ZONE** -- before the mandatory Escape encoding -- would be :
`http://sfxserver.uni.edu/sfxmenu?sid=EBSC0:MFA&id=pmid:203456&pid<author>Smith, Paul ; Klein, Calvin</author>&<yr>98</yr>`



META-TAG	VALUE	DESCRIPTION
genre	bundles:	
	journal	a journal, volume of a journal, issue of a journal
	book	a book
	conference	a publication bundling proceedings of a conference
	individual items:	
	article	a journal article
	preprint	a preprint
	proceeding	a conference proceeding
	bookitem	an item that is part of a book
aulast		A string with the first author's last name
aufirst		A string with the first author's first name
auinit		A string with the first author's first and middle initials
auinit1		A string with the first author's first initial
auinitm		A string with the first author's middle initials
issn		An ISSN number
eissn		An electronic ISSN number
coden		A CODEN
isbn		An ISBN number
sici		A SICI of a journal article, volume or issue. Compliant with ANSI/NISO Z39.56-1996 Version 2 (see http://sunsite.berkeley.edu/SICI/)
bici		A BICI for a section of a book, to which an ISBN has been assigned. Compliant with http://www.niso.org/bici.html
title		The title of a bundle (journal, book, conference)
stitle		The abbreviated title of a bundle
atitle		The title of an individual item (article, preprint, conference proceeding, part of a book)
volume		The volume of a bundle
part		The part of a bundle
issue		The issue of a bundle



spage		The start page of an individual item in a bundle
epage		The end page of an individual item in a bundle
pages		Pages covered by an individual item in a bundle. The format of this field is 'spage-epage'
artnum		The number of an individual item, in cases where there are no pages available.
date	YYYY-MM-DD YYYY-MM YYYY	The publication date of the item or bundle encoded in the "Complete date" variant of ISO8601 (see http://www.w3.org/TR/NOTE-datetime). This format is YYYY-MM-DD where YYYY is the four-digit year, MM is the month of the year between 01 (January) and 12 (December), and DD is the day of the month between 01 and 28 or 29 or 30 or 31, depending on length of the month and whether it is a leap year.
ssn	winter spring summer fall	The season of publication
quarter	1 2 3 4	The quarter of publication

