

SeeAlso Simple Specification

This is a the draft of **SeeAlso Simple Specification**. It is completed by the [SeeAlso Full Specification](#). To begin with, have a look at getting started with SeeAlso. Feedback is welcome!

Version: 0.8b (awaiting final comments)Date: 2009-02-12

- [Introduction](#)
- [Background](#)
- [SeeAlso Simple Query](#)
 - [Examples](#)
- [SeeAlso Simple Response](#)
 - [Mime Type and encoding](#)
 - [Response Format](#)
 - [Response Content](#)
 - [Identifier](#)
 - [Labels](#)
 - [Descriptions](#)
 - [URIs](#)
 - [Normalization of identifiers](#)
 - [Normalization of response content](#)
 - [Callback parameter](#)
 - [HTTP response codes](#)
- [Examples](#)
 - [Link Server](#)
 - [Search Suggestions](#)
 - [RDF Triples](#)
- [References](#)
 - [Normative](#)
 - [Non-normative](#)
- [Authors](#)
- [License](#)

Introduction

The SeeAlso Simple standard offers a convention by which web services can return a set of hyperlinks, search completions, RDF triples and similar data for a given search term. SeeAlso Simple is based on the [OpenSearch Suggestions extension 1.0](#).

Background

Among weblogs [trackback](#) and [pingback](#) are used to notify a website when it is referenced, so a backlink can be displayed. Most wikis also provide a method to get a list of backlinks to a specific page. For general information objects on the World Wide Web there is no standard API how to provide a list of links for a given context. SeeAlso Simple fills the gap by defining a simple API based on OpenSearch Suggestions. SeeAlso Simple can be used to provide link suggestions, search term suggestions, RDF triples or other simple sets of data with up to three data fields.

SeeAlso Simple Query

A SeeAlso Simple service is queried via HTTP GET. There is a single base URL for all requests. The base URL must not contain query parameters named `id` or `callback`. Each request consists of the base URL, the query parameter `id`, and a parameter value to this parameter. The query parameter `id` is also called the *search term*. In addition there can be an optional callback parameter `callback`. To allow self-describing services with [SeeAlso Full](#), it is highly recommendet to include the query parameter/value-pair `format=seealso` in the query part of the base URL.

Examples

base URL	search term	query URL	comment
http://example.com/seealso.php	1234	http://example.com/seealso.php?id=1234	simple
https://example.com:8080/	xyz:abc	https://example.com:8080/?id=xyz:abc	server on a different port
http://example.com/services?format=seealso	bar	http://example.com/services?format=seealso&id=bar	script-independent base URL including <code>format=seealso</code> (<i>recommended!</i>)

SeeAlso Simple Response

Mime Type and encoding

A SeeAlso Simple service must always return the mime type `text/javascript` or the mime type `application/x-suggestions+json`. The latter is only allowed if the response contains no callback parameter. Character encoding must be UTF-8, UTF-16, or UTF-32 [UNICODE].

Response Format

The response body must be returned in [JavaScript Object Notation](#) [JSON] as a JavaScript array of one string and three arrays of the same number of strings. A SeeAlso Simple service must not return any other format. If an error occurred, the empty response ([""], [], [], []) can be returned. In addition, the response body can be wrapped by a callback method.

Response Content

The Response is returned as an ordered collection of values. The four values are returned in the following order:

- Identifier
- Labels
- Descriptions
- URIs

The four values correspond to Query String, Completions, Descriptions, and Query URLs in the OpenSearch Suggestions Specification although their semantic can be different. Labels, Descriptions and URIs must be arrays of same length.

Identifier

The **Identifier** is a JSON string that echoes the search term, an empty string, or a normalized version of the search term (see below). It is recommended to use Uniform Resource Identifiers. The empty string should only be returned if an error occurred.

Examples:

- "urn:isbn:9780471159599"
- "info:iccn/no9910609"
- "sea"

Labels

The **Labels** are a JSON array of strings. The array must be provided, but empty strings are allowed.

Examples:

- "Bonobo", "Sue Savage-Rumbaugh", "Panzee and Panbanisha"
- "Tim Berners-Lee", "Timothy"
- "sears", "search engines", "search engine", "search", "sears.com", "seattle times"

Descriptions

The **Descriptions** are a JSON array of strings that may contain additional information about the response objects. The array must be provided, but empty strings are allowed.

Examples:

- "2007-12-02T20:19", "2007-10-14T05:41", "2007-10-13T12:55"
- "http://xmlns.com/foaf/0.1/name", "http://xmlns.com/foaf/0.1/homepage"
- "7,390,000 results", "17,900,000 results", "25,700,000 results", "1,220,000,000 results", "1 result"

URIs

The **URIs** are a JSON array of strings. Each string can either be empty or contain a Uniform Resource Identifier [URI] as defined in [RFC 3986](#).

Examples:

- "http://en.wikipedia.org/wiki/Bonobo", "http://en.wikipedia.org/wiki/Sue_Savage-Rumbaugh", "http://en.wikipedia.org/wiki/Panzee_and_Panbanisha"
- "", "", "http://www.w3.org/People/Berners-Lee"

Normalization of identifiers

Many identifiers can be presented in different notations. Therefore simple string comparison cannot be applied to determine whether two identifiers are equal. A frequently neglected solution to this problem is to normalize identifiers to a canonical form. SeeAlso recommends services to normalize identifiers in the response content. This standard does not define whether and how to do normalization, but if a SeeAlso Simple service provides normalization of identifiers, the response must be equal for all non-normalized identifiers that are normalized to the same canonical form (apart from general changes over time). See [RFC3986, section 6](#) for more about normalization of URIs.

Examples:

- International Standard Book Numbers [ISBN] are used to uniquely identify monographs since the 1970s. Since 2007 an ISBN is a 13-digit number (ISBN-13). Until then 10-digit numbers (ISBN-10) had been used. A ISBN-10 can be converted to ISBN-13 but not vice versa. ISBN can be written with additional hyphens to separate parts. [RFC 3187](#) defines the `urn:isbn` namespace to use ISBNs as Uniform resource names (URN). So a canonical form of ISBN could be defined as the ISBN-13 without hyphens and prefixed with `urn:isbn:`. The normalization defines unique URIs for ISBNs. A SeeAlso Simple service could support different notations of an ISBN but always return the normalized form as URI.

- The Library of Congress identifies cataloging records with Library of Congress Control Numbers (LCCN). A method to normalize LCCN to canonical form is defined in [LCCN-NS](#). The info-URI scheme [INFO-URI] contains a namespace for LCCN. A SeeAlso Simple service could use the info-URI as the normalized version of a LCCN. It could then assure to understand different forms of a LCCN like "no99-10609", "LCCN: no9910609", and "info:lccn/no9910609" and normalize them to the canonical form "info:lccn/no9910609".

If a SeeAlso Simple service provides normalization of identifiers, then it must return normalized identifiers also if the rest of the response content is empty. For instance a service that uses ISBN as search terms and provides normalization, should return ["urn:isbn:9780471159599"].[],[] (instead of ["", [], []] if the search term is "978-0-471-15959-9 and no response labels, descriptions, and URIs were found.

Normalization of response content

A response may contain empty strings for values in the Labels, Descriptions, and URIs. If all of them are empty strings for a given array index, the response can be normalized by removing these empty elements.

Examples:

- The response ["abc",["a",""],["b",""],["",""]] can be normalized to ["abc",["a"],["b"],[""]]
- The response ["x",["",""],["",""],["",""]] is equal to the empty response ["x",[],[]].

Callback parameter

A SeeAlsoSimple service can support an additional callback parameter (`callback=function`). The callback parameter wraps the JSON output text in parentheses and a function name of your choosing. Callback function names may only use upper and lowercase alphabetic characters (A-Z, a-z), numbers (0-9), the period (.), the underscore (_), and brackets ([and]). Brackets must be URL-encoded.

Examples:

- Query: http://example.com/myservice?id=123&callback=show_links
Response: `show_links(...json output...);`
- Query: <http://example.com/myservice?id=123&callback=links.show%5B2%5D>
Response: `links.show[2](...json output...);`

HTTP response codes

To support AJAX calls with dynamic script tags, a SeeAlso Simple service must return HTTP response code "200 - OK" [HTTP] for all valid responses no matter whether the response content is empty or not. If a service supports callback parameter, the error code "400 - Bad Request" can be returned if a malformed callback parameter value was given.

Examples

Link Server

Many articles of the collaborative encyclopedia Wikipedia contain references to publications with International Standard Book Numbers [ISBN]. You could provide a SeeAlso Simple Service with base URL <http://example.com/seealso/isbn2wikipedia?lang=en> that gets an ISBN and provides a list of links to English Wikipedia articles that contain this ISBN. Because a ISBN can occur in multiple variants (with or without hyphens, ISBN-10 or ISBN-13), the webservice should normalize it to a canonical form. A sample query to the link server would be

<http://example.com/seealso/isbn2wikipedia?lang=en&id=0-471-15959-X>

A matching response (with additional spaces for better readability) would be:

```
[ "urn:isbn:9780471159599",
  [ "Bonobo", "Sue Savage-Rumbaugh", "Panzee and Panbanisha" ],
  [ "2007-12-02T20:19", "2007-10-14T05:41", "2007-10-13T12:55" ],
  [ "http://en.wikipedia.org/wiki/Bonobo"
    "http://en.wikipedia.org/wiki/Sue_Savage-Rumbaugh",
    "http://en.wikipedia.org/wiki/Panzee_and_Panbanisha" ]
]
```

In this example the Descriptions of the response contain the time when an article was last modified. The ISBN is normalized to ISBN-13 without hyphens and prefixed with `urn:isbn:`. Please note that <http://example.com/seealso/isbn2wikipedia> is not a SeeAlso Simple base URL in this example. Further services could be provided for any other Wikipedia language XY at <http://example.com/seealso/isbn2wikipedia?lang=XY> - but each of them would be an independent SeeAlso Simple Service according to this specification.

Search Suggestions

All valid OpenSearch Suggestions responses are also valid SeeAlso Simple responses. The following is a full example of a JSON formatted SeeAlso Simple response that encodes suggested search queries:

```
[ "sea",
  [ "sears", "search engines", "search engine", "search", "sears.com" ],
  [ "7,390,000 results", "17,900,000 results", "25,700,000 results", "1,220,000,000 results", "1 result" ],
  [ "http://example.com?q=sears", "http://example.com?q=search+engines",
    "http://example.com?q=search+engine", "http://example.com?q=search", "http://example.com?q=sears.com" ]
]
```

RDF Triples

SeeAlso Simple can be used as a simple API to deliver RDF triples with a common subject. To encode RDF in a SeeAlso response, the Identifier contains the RDF triple's subject, the Description contains the RDF triple's predicate, and the RDF triple's object is contained in the Label (literal) or URI (RDF URI reference).

For instance two simple statements about a specific entry in the Library of Congress Authorities can be formulated in Notation 3 [N3] this way:

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix lccn: <info:lccn/> .

lccn:no9910609
  foaf:name "Tim Berners-Lee";
  foaf:firstName "Timothy";
  foaf:homepage <http://www.w3.org/People/Berners-Lee>.
```

The SeeAlso Simple Response to encode this triples is:

```
[ "info:lccn/no9910609",
  [ "Tim Berners-Lee",
    [ "http://xmlns.com/foaf/0.1/name", "http://xmlns.com/foaf/0.1/firstName", "http://xmlns.com/foaf/0.1/homepage"
  ]
  [ "",
    [ "http://www.w3.org/People/Berners-Lee" ]
  ]
]
```

Please note that literal values with language tag or data types (typed literals) are not supported. Literals are always plain literals without language tag.

If for a given response element both URI and label are provided, then the label has no semantics but may only be of practical value (for instance for display). For example the Response ["A","B","C","D"] encodes the RDF triple (in N3) A C D. instead of A C "B".

References

Normative

- **[HTTP]** [Hypertext Transfer Protocol - HTTP/1.1](#). June 1999 (=RFC 2616)
- **[JSON]** [JavaScript Object Notation](#). (=RFC 4627)
- **[URI]** [Uniform Resource Identifiers \(URI\): Generic Syntax](#). August 1998. (=RFC 2396)
- **[UNICODE]** [The Unicode Standard Version 5.0](#) The Unicode Consortium, 2007

Non-normative

- **[FOAF]** [FOAF Vocabulary Specification 0.91](#). November 2007.
- **[INFO-URI]** [The "info" URI Scheme for Information Assets with Identifiers in Public Namespaces](#). April 2006. (=RFC 4452)
- **[ISBN]** [International standard book number \(ISBN\)](#). (=ISO 2108:2005)
- **[LCCN-NS]** [The Library of Congress Control Number Namespace](#). Library Of Congress, November 2003.
- **[N3]** [Notation 3](#). Tim Berners-Lee (Editor)
- **[OSS]** [OpenSearch Suggestions 1.0](#). Public draft, September 2006.
- **[RDF-PRIMER]** [RDF Primer](#). W3C Working Draft, November 2002.
- **[URI:ISBN]** [Using International Standard Book Numbers as Uniform Resource Names](#). October 2001 (=RFC 3187)

Authors

This document was created by Jakob Voss <jakob.voss@gbv.de> based on the OpenSearch Suggestions 1.0 Specification. The OpenSearch Suggestions 1.0 documentation was written up by DeWitt Clinton <dewitt@opensearch.org>, based directly on the work of Joe Hughes' Firefox search suggestions draft documentation and the Google search suggestions format.

License

This document is made available subject to the terms of the [Creative Commons Attribution-ShareAlike 2.5 License](#).