Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 10, 2008

J. Gregorio, Ed.

M. Hadley, Ed.
Sun Microsystems
M. Nottingham, Ed.

D. Orchard
BEA Systems, Inc.
July 9, 2007

URI Template
draft-gregorio-uritemplate-01

Status of this Memo

Copyright Notice

Abstract

URI Templates are strings that can be transformed into URIs after embedded variables are substituted. This document defines the syntax and processing of URI Templates.

Editorial Note

To provide feedback on this Internet-Draft, join the W3C URI mailing list (http://lists.w3.org/Archives/Public/uri/) [1].

Table of Contents

1. Introduction

URI Templates are strings that contain embedded variables that are transformed into URIs after embedded variables are substituted.

This is useful when it's necessary to convey the structure of a URI in a well-defined way. For example, documentation of an interface exposed by a Web site might use a template to show people how to find information about a user;

    http://www.example.com/users/{userid}

URI Templates can also be thought of as the basis of a machine-readable forms language; by allowing clients to form their own identifiers based on templates given to them by the URI's authority, it's possible to construct dynamic systems that use more of the URI than traditional HTML forms are able to. For example,

    http://www.example.org/products/{upc}/buyers?page={page_num}

Finally, URI Templates can be used to compose URI-centric protocols without impinging on authorities' control of their URIs. For example, there are many emerging conventions for passing around login information between sites using URIs. Forcing people to use a well-known query parameter isn't good practice, but using a URI parameter allows different sites to specify local ways of conveying the same information;

    http://login.example.org/login?back={return-uri}
    http://auth.example.com/userauth;{return-uri}

This specification defines the basic syntax and processing of URI Templates. Each application of URI Templates will need to define its own profile of this specification that indicates what template variables are available, how to convey them to clients, and what their appropriate use is in that context.

2. Notational Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

This specification uses the Augmented Backus-Naur Form (ABNF) notation of [RFC4234]. See [RFC3986] for the definitions of the URI-reference, reserved, and unreserved rules.

3. URI Template

A URI Template is a sequence of characters that contains one or more embedded template variables, see Section 3.1. A URI Template becomes a URI when the template variables are substituted with their values (see Section 3.2). For example:

http://example.com/widgets/{widget_id}

If the value of the widget_id variable is "xyzzy", the resulting URI after substitution is:

http://example.com/widgets/xyzzy

3.1. Template Variables

Template variables are the parameterized components of a URI Template. A template variable MUST match the template-var rule.

```
template-char = unreserved
template-name  = 1*template-char
template-var  = "{" template-name "}"
```

3.2. URI Template Substitution

Evaluating a URI Template ("substitution") consists of replacing all template variables with their respective string values.

During substitution, the string value of a template variable MUST have any characters that do not match the reserved or unreserved rules (i.e., those characters not legal in URIs without percent encoding) percent-encoded, as per [RFC3986], section 2.1. Specific applications of URI Templates MAY specify additional constraints and encoding rules in addition to this.

Any number of template variables MAY appear in a URI Template; a single template-name MAY appear multiple times.

The result of substitution MUST match the URI-reference rule and SHOULD also match any known rules for the scheme of the resulting URI.

Typically, this is ensured by the definitions of the template variables used. For example, they may specify that a variable's value is not to contain certain characters, or that some characters should be percent-encoded before substitution.

3.3. Using URI Templates

Applications using URI Templates will typically need to specify a number of things, including;

- The template to use.
- What template variables are available.
- For each of the variables;
  - What characters are allowed in the template's value.
  - What encodings should be applied to the value before substitutions.
  - How to handle errors such as the output of substitution being an invalid URI.

URI Template processors SHOULD allow applications to indicate that;

- A variable's value is required to contain at least one character
- A variable's value is required to match one of a set of supplied options
- A variable's value is to have all reserved characters, as per RFC3986, percent-escaped before substitution

Processors MAY make additional options available.

3.3.1. Examples

Given the following template names and values:

Name	Value
a	fred
b	barney
c	cheeseburger
d	one two three
e	20% tricky
f	
20	this-is-spinal-tap
scheme	https
p	quote=to+be+or+not+to+be
q	hullo#world

Table 1

(Note that the name 'wilma' has not been defined, and the value of 'f' is the empty string.)

The following URI Templates will be expanded as shown:

```
http://example.org/page1#{a}
http://example.org/page1#fred

http://example.org/{a}/{b}/
http://example.org/fred/barney/

http://example.org/{a}{b}/
http://example.org/fredbarney/

http://example.com/order/{c}/{c}/{c}/
http://example.com/order/cheeseburger/cheeseburger/cheeseburger/

http://example.org/{d}
http://example.org/one%20two%20three

http://example.org/{e}
http://example.org/20%25%20tricky

http://example.com/{f}/
http://example.com//

{scheme}://{20}.example.org?date={wilma}&option={a}
https://this-is-spinal-tap.example.org?date=&option=fred

http://example.org?{p}
http://example.org?quote=to+be+or+not+to+be

http://example.com/{q}
http://example.com/hullo#world
```

4. Security Considerations

A URI Template does not contain active or executable content. Other security considerations are the same as those for URIs, see section 7 of RFC3986.

5. IANA Considerations

In common with RFC3986, URI scheme names form a registered namespace that is managed by IANA according to the procedures defined in [RFC4395]. No IANA actions are required by this document.

6. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005.

[RFC4234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 4234, October 2005.

[RFC4395] Hansen, T., Hardie, T., and L. Masinter, "Guidelines and Registration Procedures for New URI Schemes", BCP 115, RFC 4395, February 2006.

[1] <http://lists.w3.org/Archives/Public/uri/>

Appendix A. Contributors

The following people made significant contributions to this specification: DeWitt Clinton and James Snell.

Appendix B. Revision History

01

00 - Initial Revision.

Authors' Addresses

Joe Gregorio (editor)

Email: joe@bitworking.org
URI: http://bitworking.org/

Marc Hadley (editor)
Sun Microsystems

Email: Marc.Hadley@sun.com
URI: http://sun.com/

Mark Nottingham (editor)

Email: mnot@pobox.com
URI: http://mnot.net/

David Orchard
BEA Systems, Inc.

Email: dorchard@bea.com
URI: http://bea.com/

Full Copyright Statement

Intellectual Property

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).