

appsawg
Internet-Draft
Updates: [3986](#) (if approved)
Intended status: BCP
Expires: August 3, 2014

M. Nottingham
January 30, 2014

URI Design and Ownership
draft-ietf-appsawg-uri-get-off-my-lawn-01

Abstract

Sometimes, it is attractive to add features to protocols or applications by specifying a particular structure for URIs (or parts thereof). However, publishing standards that mandate URI structure is inappropriate because the structure of a URI needs to be firmly under the control of its owner, and the IETF (as well as other organisations) should not usurp this ownership.

This document is intended to prevent this practice (sometimes called "URI Squatting") in standards, but updating [RFC3986](#) to indicate where it is acceptable.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 3, 2014.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of

publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	3
1.1.	Who This Document Is For	4
1.2.	Notational Conventions	4
2.	Best Current Practices for Standardising Structured URIs	5
2.1.	URI Schemes	5
2.2.	URI Authorities	5
2.3.	URI Paths	5
2.4.	URI Queries	5
2.5.	URI Fragment Identifiers	6
3.	Security Considerations	6
4.	IANA Considerations	6
5.	References	6
5.1.	Normative References	6
5.2.	Informative References	7
Appendix A.	Acknowledgments	7
Appendix B.	Alternatives to Specifying Structure in URIs	7
	Author's Address	8

1. Introduction

URIs [[RFC3986](#)] very often include structured application data. This might include artifacts from filesystems (often occurring in the path component), and user information (often in the query component). In some cases, there can even be application-specific data in the authority component (e.g., some applications are spread across several hostnames to enable a form of partitioning or dispatch).

Furthermore, constraints upon the structure of URIs can be imposed by an implementation; for example, many Web servers use the filename extension of the last path segment to determine the media type of the response. Likewise, pre-packaged applications often have highly structured URIs that can only be changed in limited ways (often, just the hostname and port they are deployed upon).

Because the owner of the URI is choosing to use the server or the software, this can be seen as reasonable delegation of authority. When such conventions are mandated by standards, however, it can have several potentially detrimental effects:

- o Collisions - As more conventions for URI structure become standardised, it becomes more likely that there will be collisions between such conventions (especially considering that servers, applications and individual deployments will have their own conventions).
- o Dilution - When the information added to a URI is ephemeral, this dilutes its utility by reducing its stability (see [[webarch](#)] [Section 3.5.1](#)), and can cause several alternate forms of the URI to exist (see [[webarch](#)] [Section 2.3.1](#)).
- o Rigidity - Fixed URI syntax often interferes with desired deployment patterns. For example, if an authority wishes to offer several applications on a single hostname, it becomes difficult to impossible to do if their URIs do not allow the required flexibility.
- o Operational Difficulty - Supporting some URI conventions can be difficult in some implementations. For example, specifying that a particular query parameter be used precludes the use of Web servers that serve the response from a filesystem. Likewise, an application that fixes a base path for its operation (e.g., "/v1") makes it impossible to deploy other applications with the same prefix on the same host.
- o Client Assumptions - When conventions are standardised, some clients will inevitably assume that the standards are in use when those conventions are seen. This can lead to interoperability problems; for example, if a specification documents that the "sig" URI query parameter indicates that its payload is a cryptographic signature for the URI, it can lead to undesirable behaviour.

Nottingham

Expires August 3, 2014

[Page 3]

While it is not ideal when a server or a deployed application constrains URI structure (indeed, this is not recommended practice, but that discussion is out of scope for this document), publishing standards that mandate URI structure (beyond those allowed by [\[RFC3986\]](#)) is inappropriate because the structure of a URI needs to be firmly under the control of its owner, and the IETF (as well as other organisations) should not usurp this ownership; see [\[webarch\]](#) [Section 2.2.2.1](#).

This document explains best current practices for establishing URI structures, conventions and formats in standards. It also offers strategies for specifications to avoid violating these guidelines in [Appendix B](#).

[1.1](#). Who This Document Is For

This document's requirements specifically target a few different types of specifications:

- o URI Scheme Definitions ("scheme definitions") - specifications that define and register URI schemes, as per [\[RFC4395\]](#).
- o Protocol Extensions ("extensions") - specifications that offer new capabilities to potentially any identifier, or a large subset; e.g., a new signature mechanism for 'http' URIs, or metadata for any URI.
- o Applications Using URIs ("applications") - specifications that use URIs to meet specific needs; e.g., a HTTP interface to particular information on a host.

Requirements that target the generic class "Specifications" apply to all specifications, including both those enumerated above and others.

Note that this specification ought not be interpreted as preventing the allocation of control of URIs by parties that legitimately own them, or have delegated that ownership; for example, a specification might legitimately define the semantics of a URI on the IANA.ORG Web site as part of the establishment of a registry.

[1.2](#). 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 [\[RFC2119\]](#).

Nottingham

Expires August 3, 2014

[Page 4]

2. Best Current Practices for Standardising Structured URIs

Best practices differ depending on the URI component.

2.1. URI Schemes

Applications and extensions MAY require use of specific URI scheme(s); for example, it is perfectly acceptable to require that an application support 'http' and 'https' URIs. However, applications SHOULD NOT preclude the use of other URI schemes in the future, unless they are clearly specific to the nominated schemes.

A specification that defines substructure within a URI scheme MUST do so in a registration document for the URI scheme in question, or by modifying [[RFC4395](#)].

2.2. URI Authorities

Scheme definitions define the presence, format and semantics of an authority component in URIs; all other specifications MUST NOT constrain, define structure or semantics for URI authorities.

For example, an extension or application cannot say that the "foo" prefix in "foo_app.example.com" is meaningful or triggers special handling.

2.3. URI Paths

Scheme definitions define the presence, format, and semantics of a path component in URIs; all other specifications MUST NOT constrain, define structure or semantics for any path component.

The only exception to this requirement is registered "well-known" URIs, as specified by [[RFC5785](#)]. See that document for a description of the applicability of that mechanism.

For example, an application cannot specify a fixed URI path "/myapp", since this usurps the host's control of that space. Specifying a fixed path relative to another (e.g., {whatever}/myapp) is also bad practice, since it "locks" the URIs in use; while doing so might prevent collisions, it does not avoid the other issues discussed.

2.4. URI Queries

The presence, format and semantics of the query component of URIs is dependent upon many factors, and MAY be constrained by a scheme definition. Often, they are determined by the implementation of a resource itself.

Applications SHOULD NOT directly specify the syntax of queries, as this can cause operational difficulties for deployments that do not support a particular form of a query.

Extensions MUST NOT specify the format or semantics of queries.

For example, an extension cannot be minted that indicates that all query parameters with the name "sig" indicate a cryptographic signature.

2.5. URI Fragment Identifiers

Media type definitions (as per [[RFC6838](#)]) SHOULD specify the fragment identifier syntax(es) to be used with them; other specifications MUST NOT define structure within the fragment identifier, unless they are explicitly defining one for reuse by media type definitions.

3. Security Considerations

This document does not introduce new protocol artifacts with security considerations. It prohibits some practices that might lead to vulnerabilities; for example, if a security-sensitive mechanism is introduced by assuming that a URI path component or query string has a particular meaning, false positives might be encountered (due to sites that already use the chosen string).

4. IANA Considerations

This document clarifies appropriate registry policy for new URI schemes, and potentially for the creation of new URI-related registries, if they attempt to mandate structure within URIs. There are no direct IANA actions specified in this document.

5. References

5.1. 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.
- [RFC4395] Hansen, T., Hardie, T., and L. Masinter, "Guidelines and

Registration Procedures for New URI Schemes", [BCP 35](#), [RFC 4395](#), February 2006.

- [RFC6838] Freed, N., Klensin, J., and T. Hansen, "Media Type Specifications and Registration Procedures", [BCP 13](#), [RFC 6838](#), January 2013.

[5.2.](#) Informative References

- [RFC5785] Nottingham, M. and E. Hammer-Lahav, "Defining Well-Known Uniform Resource Identifiers (URIs)", [RFC 5785](#), April 2010.
- [RFC5988] Nottingham, M., "Web Linking", [RFC 5988](#), October 2010.
- [RFC6570] Gregorio, J., Fielding, R., Hadley, M., Nottingham, M., and D. Orchard, "URI Template", [RFC 6570](#), March 2012.
- [webarch] Jacobs, I. and N. Walsh, "Architecture of the World Wide Web, Volume One", December 2004, <<http://www.w3.org/TR/2004/REC-webarch-20041215>>.

[Appendix A.](#) Acknowledgments

Thanks to David Booth, Dave Crocker, Tim Bray, Anne van Kesteren, Martin Thomson and Erik Wilde for their suggestions and feedback.

[Appendix B.](#) Alternatives to Specifying Structure in URIs

Given the issues above, the most successful strategy for applications and extensions that wish to use URIs is to use them in the fashion they were designed; as links that are exchanged as part of the protocol, rather than statically specified syntax. Several existing specifications can aid in this.

[RFC5988] specifies relation types for Web links. By providing a framework for linking on the Web, where every link has a relation type, context and target, it allows applications to define a link's semantics and connectivity.

[RFC6570] provides a standard syntax for URI Templates that can be used to dynamically insert application-specific variables into a URI to enable such applications while avoiding impinging upon URI owners' control of them.

[RFC5785] allows specific paths to be 'reserved' for standard use on

URI schemes that opt into that mechanism ('http' and 'https' by default). Note, however, that this is not a general "escape valve" for applications that need structured URIs; see that specification for more information.

Specifying more elaborate structures in an attempt to avoid collisions is not adequate to conform to this document. For example, prefixing query parameters with "myapp_" does not help, because the prefix itself is subject to the risk of collision (since it is not "reserved").

Author's Address

Mark Nottingham

Email: mnot@mnot.net

URI: <http://www.mnot.net/>

