Network Working Group
Internet-Draft
Updates: 3986 (if approved)
Intended status: BCP
Expires: February 3, 2014

M. Nottingham
August 2, 2013

Standardising Structure in URIs
draft-nottingham-uri-get-off-my-lawn-00

Abstract

It is sometimes attractive to specify a particular structure for URIs (or parts thereof) to add support for a new feature, application or facility. This memo provides guidelines for such situations in standards documents.

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 February 3, 2014.

Table of Contents

1. Introduction

URIs [RFC3986] very often include structure and application data. This might include artefacts from filesystems (often occuring in the path component), user information (often in the query component) and application data throughout. 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).

Such conventions for 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).

When such conventions are mandated by standards, however, it can have several potentially detrimental effects:

- 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).
- Dilution - Adorning URIs with extra information to support new standard features dilutes their usefulness as identifiers when that information is ephemeral (as URIs ought to be stable; see [webarch] Section 3.5.1), or its inclusion causes several alternate forms of the URI to exist (see [webarch] Section 2.3.1).
- Operational Difficulty - Supporting some URI conventions can be difficult in some implementations. For example, specifying that a particular query parameter be used preclude the use of Web servers that serve the response from a filesystem.
- Client Assumptions - When conventions are standardised, some clients will inevitably assume that the standards are in use when they are seen. This can lead to interoperability problems.

At a more philosophical level, the structure of a URI needs to be firmly under the control of a single party; its owner. Standardising parts of a URI's structure usurps that control; see [webarch] Section 2.2.2.1 for more information.

This memo explains best current practices for establishing URI structures, conventions and formats in specifications; in particular, IETF specifications, although they are more broadly applicable. It also offers strategies for specifications to avoid violating these guidelines in Section 3.

1.1. 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].

2. Best Current Practices for Standarising Structured URIs

These guidelines target a few different types of specifications:

- URI Scheme Definitions ("scheme definitions") - specifications that define and register URI schemes, as per [RFC4395].
- 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.
- 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 standards, including both those enumerated above above and others. They are also applicable to non-standard RFC publications.

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 specify the semantics of a URI on the IANA.ORG Web site as part of the establishment of a registry.

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, to promote reuse, unless they are clearly specific to the nominated schemes.

Specifications MUST NOT define substructure within URI schemes, unless they do so by modifying [RFC4395], or they are the registration document for the URI scheme(s) in question.

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 them.

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 them.

The only exception to this requirement is registered "well-known" URIs, as specified by [RFC5785].

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. In particular, extensions MUST NOT assume that all HTTP(S) resources are capable of accepting queries in the format defined by [HTML4], Section 17.13.4.

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. Alternatives to Specifying Static 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 run-time artefacts that are exchanged as part of the protocol, rather than staticly specified syntax.

For example, if a specific URI needs to be known to interact with an application, its "shape" can be determined by interacting with the application's more general interface (in Web terms, its "home page") to learn about that URI.

[RFC5988] describes a framework for identifying the semantics of a

link in a "link relation type" to aid this. [RFC6570] provides a standard syntax for "link 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.

4. Security Considerations

This memo does not introduce new protocol artefacts with security considerations.

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.

[RFC5785] Nottingham, M. and E. Hammer-Lahav, "Defining Well-Known Uniform Resource Identifiers (URIs)", RFC 5785, April 2010.

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

5.2. Informative References

[HTML4] Jacobs, I., Le Hors, A., and D. Raggett, "HTML 4.01 Specification", December 1999, <http://www.w3.org/TR/REC-html40/>.

[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>.

Author's Address

Mark Nottingham

Email: mnot@mnot.net
URI: http://www.mnot.net/