The use of Metadata in URIs

DRAFT TAG Finding 09 June 2006

This version:
http://www.w3.org/2001/tag/doc/metaDataInURI-31-20060609.html
Latest version:
http://www.w3.org/2001/tag/doc/metaDataInURI-31 (XML)
Previous versions:
Unapproved Editors Drafts:
http://www.w3.org/2001/tag/doc/metaDataInURI-31-20060511.html,
http://www.w3.org/2001/tag/doc/metaDataInURI-31-20030708.html,
http://www.w3.org/2001/tag/doc/metaDataInURI-31-20030704.html
(W3C Member-only)
Editors:
Noah Mendelsohn <noah_mendelsohn@us.ibm.com>
Stuart Williams <skw@hplb.hpl.hp.com>

Abstract

Status of this Document

Editors DRAFT

This document has been developed for discussion by the W3C Technical Architecture Group. This finding addresses the TAG issue metadataInURI-31.

The content of this document is intended for discussion and does NOT necessarily represent a consensus position of the TAG. An informal guide to previous discussion of this topic is available and may be useful to reviewers of this draft.

The terms MUST, MUST NOT, SHOULD, and SHOULD NOT are used in this document in accordance with [RFC2119].

Publication of this finding does not imply endorsement by the W3C Membership. This is a draft document and may be updated, replaced or obsoleted by other documents at any time.

Additional TAG findings, both approved and in draft state, may also be available.

Please send comments on this finding to the publicly archived TAG mailing list www-tag@w3.org (archive).

Table of Contents

---

1 Introduction

Web-based software uses URIs to designate resources for retrieval or for other operations. The authority that creates a URI is responsible for assuring that it is associated with the intended resource, and that the appropriate data is manipulated or returned in response to operations that use the URI as a resource designator. Many URI schemes offer a flexible structure that can also be used to carry additional information, called metadata, about the resource. Such metadata might include the title of a document, the creation date of the resource, the MIME media type that is likely to be returned by an HTTP GET, a digital signature usable to verify the integrity or authorship of the resource content, or hints about URI assignment policies that would allow one to guess the URIs for related resources.

Comment [skw1]: The concept of authority wrt to URI is one which some have pushed back against. They have argued that the URI scheme itself is what states what a given URI identifies. Generally this is presented as an operationalised notion of what it means to 'identify' a resource. This view would likely also argue that RFC2616 'creates' all possible HTTP URIs.

Comment [skw2]: Do you have an example of such a scheme. I can't think of any!!!

This finding addresses several questions regarding such metadata in URIs:

1. What information about a resource can or should be embedded in its URI?
2. What metadata can be *reliably* determined from a URI, and in what circumstances is it appropriate to rely on the correctness of such information?
3. In what circumstances is it appropriate to use information from a URI as a hint as to the nature of a resource or its representations?

The first question is focused on people and software acting in the role of or on behalf of a URI assignment authority (authorities) for URI assignments within

the scope of that authority. The other questions are focused on people and software making use of URIs assigned outside of their own authority (observers). Of course, the questions are related, insofar is one reason for an authority to encode metadata is for the benefit of resource users.

Comment [skw3]: Whilst I'm conscious that this is either text that I wrote or similar, it is again couched in terms of authority, which I know some rejects. That said I think that there may be a crossing of layers here in that an operationalised view of what a given URI identifies has nothing to say about what a resource signifies.

FWIW IIRC Roy on the other hand supported the notion of delegated authority passed on downward from the URI spec to scheme specs, to 'owners' of DNS names and so forth.

The TAG has earlier published a finding *Authoritative Metadata* [AUTHMETA], which explains how to determine correct metadata in cases where conflicting information has been provided. This finding is concerned with just one possible means of determining resource metadata, i.e. from the URI itself.

2 Encoding and using metadata in URIs

This section uses simple examples to illustrate some issues that arise when encoding metadata in URIs, or when relying on information gleaned from such URIs. *Good Practice Notes* are provided to explain how to use the Web effectively, and *Constraints* are given where necessary for using the Web correctly. As these examples show, encoding or not encoding metadata in a URI or deciding whether to rely on such metadata is often a tradeoff, involving some benefits and some costs. In such cases, choices should be made that best meet the needs of particular resource providers and users.

2.1 Reliability of URI metadata

Consider Martin, who is using a Web-based bug tracking system to investigate some software problems. He sees a bug report which says:

> "See http://example.org/bugdata/brokenfile.xml for an example of XML that is not well-formed."

The bug tracking system is built to show examples just as they are entered into the system, so for http://example.org/bugdata/brokenfile.xml it returns a stream of (poorly formed) XML with Content-Type text/plain. That Content-Type should cause a properly configured browser to show Martin the erroneous text just as it was recorded:

```
<?xml version="1.0">
<PetList>
  <Dog>Rover</Dog>
  <Cat>Felix</Fish>
</PetList>
```

Unfortunately, Martin uses a browser that incorrectly attempts to infer the format of the returned data from the URI suffix. Keying on the ".xml" in the URI, it launches an XML renderer for what should have been plain text. When Martin attempts to view the faulty file, he sees instead a browser error saying that the erroneous XML could not be displayed.

Constraint

Constraint: Web software MUST NOT depend on the correctness of metadata inferred from a URI, except as licensed by applicable standards and specifications.

Such standards and specifications include pertinent Web and Internet RFCs and Recommendations such as [URI], as well as documentation provided by the URI assignment authority.

In this example, there is no normative specification that provides for determination of a media-type from URI suffixes, and the assignment authority has provided no documentation to license an inference of media-type from the URI. Martin's browser is in error, because it relies on URI metadata that is not covered by normative specifications and has not been documented by the assignment authority. A correctly written browser would have shown the faulty XML as text, or might conceivably have shown a warning about the apparent mismatch between the type inferred from the URI and the returned Content-Type. (Martin's browser is also ignoring TAG finding "Authoritative Metadata" [AUTHMETA], which mandates that the Content-Type HTTP header takes precedence even if type information had somehow been reliably encoded in the URI.)

Comment [skw4]: It is in error because it **construes** that there is metadata intentionally placed in the URI when there is not.

Note that the constraint refers to *conclusions* drawn by software, which must be trustworthy, as opposed to guesses made by people. As discussed in **2.3 Guessing information from a URI**, guessing is something that people using the Web do quite often and for good reason. Software tends to be long lived and widely distributed. Thus unlicensed metadata dependencies in software result not only in buggy systems, but in inappropriate expectations that authorities will constrain their URI assignment policies and representation types to match the dependencies in the clients. For both of these reasons, the constraint above requires that software must not have such unlicensed dependencies.

Deleted: reaons

There is certain metadata that Martin or his browser can reliably determine from the URI. For example, the URI conveys that the http scheme has been used, and that attempts to access the resource should be directed to the IP address returned from the DNS resolution of the string "example.org". These conclusions are licensed by normative specifications such as [URI] and [HTTP].

Comment [skw5]: Hmmmm I have always found this tricky. Wrt to say FTP URI scheme, the scheme tells you (in an operational style) what resource is identified – it is the resource that would provide the resulting representation *if* you did a particular bunch of things. The HTTP spec is the same. However, neither is a statement about HOW the resource should be accessed, only a statement of WHAT resource is identified. Ok. Yes, typically HTTP: would imply that access using http ought to be possible.

2.2 Avoid depending on metadata

There is almost always a cost to peeking into a URI to get metadata. Even when Web architecture and the guidelines above say that you *may* do so, you should be reluctant, especially when constructing general purpose Web software. Software that peeks is less likely to work with arbitrary resources than software that doesn't. For example, software that works only with URIs in the http scheme is less general than software that works for arbitrary URIs. Software that attempts to act on "file extension" suffixes, such as .jpeg, is likely to be doing so in violation of Web Architecture, and in any case such software won't work with URIs that don't have the suffix. Even at the

assignment authority, which has definitive knowledge of the metadata encoded in its URIs, software that's dependent on such encodings will only be usable for resources that obey the convention.

Good Practice

Good Practice: Avoid software dependencies on metadata in URIs.

Comment [skw6]: The tone of this seems to me to have a presumption that metadata *is* embedded in URIs, as opposed to "in some cases there happens to be metadata embedded in URIs".

I find myself not wanting to allow that the things being cited here as metadata are infact metadata. I see them mostly as 'distinguishing' characteristics which have been encoded into URIs principally for the purpose of generating unique, transcribable URIs, rather than with the intent that metadata be recoverable from the URI.

2.3 Guessing information from a URI

Bob is walking down a street, and he sees an advertisement on the side of a bus:

"For the best Chicago Weather information on the Web, visit http://example.org/weather/Chicago."

Bob goes home, and types the URI into his browser, which does indeed display for him a Chicago weather forecast. Bob then realizes that he'll be visiting Boston, and he guesses that a Boston weather page might be available at a similar URI:

Bob guesses the Boston weather might be found at "http://example.org/weather/Boston".

He types that into his browser and reads the response that comes back.

Bob is using the original URI for more than its intended purpose, which is to identify the Chicago weather page. Instead, he's inferring from it information about the structure of a Web site that, he guesses, might use a uniform naming convention for the weather in lots of cities. So, when Bob tries the Boston URI, he has to be prepared for the possibility that his guess will prove wrong: Web architecture does not guarantee that the retrieved page, if there is one, has the weather for Boston, or indeed that it contains any weather report at all. Even if it does, there is no assurance that it is current weather, that it is intended for reliable use by consumers, etc. Bob has seen an advertisement listing just the Chicago URI, and that is the only one for which the URI authority has taken specific responsibility.

Comment [skw7]: Hmmm… I might argue that the same assignment authority is equally *responsible* for both URIs, however they have set no particular expectation wrt to the second URI (at least in the vicinity of Chicago – though who knows what might happen to be painted on the side of busses in Boston).

Still, the ability to explore the Web informally and experimentally is very valuable, and Web users act on guesses about URIs all the time. Many authorities facilitate such flexible use of the Web by assigning URIs in an orderly and predictable manner. Nonetheless, in the example above, Bob is responsible for determining whether the information returned is indeed what he needs.

Good Practice

Good Practice: Guess information from URIs only when the consequences of an incorrect guess are acceptable.

Comment [skw8]: Alternative formulation: "When guessing information from URIs be robust to unexpected results."

2.4 HTML Forms, and Documenting Metadata Assignment Policies

Bob would not have had to guess the Boston weather URI if the authority had documented its URI assignment policy. Assignment authorities have no obligation to provide such documentation, but it can be a useful way of advertising in bulk the URIs for a collection of related resources. For example, the advertisement might have read:

> "For the best weather information for your city, visit http://example.org/weather/*your-city-name-here*."

Reading that advertisement, Bob can reasonably assume that weather reports are available by substituting specific city names into the URI pattern `http://example.org/weather/your-city-name-here`. Moreover, the advertisement claims that the weather information obtainable at those URIs is "the best", so Bob can assume that the weather reports are trustworthy and current.

HTML forms [HTMLForms] and now XForms [XFORMS] each provide a means by which an authority can assert its support for a class of parameterized URIs, while simultaneously programming Web clients to prompt for the necessary parameters. For example, a Web site `http://example.org/weatherfinder` might offer a city lookup page containing the following HTML form fragment:

```
<FORM ACTION="http://example.org/cityweather" METHOD="GET">
  For what city would you like a weather report: <INPUT TYPE="TEXT"
NAME="city">?
  <INPUT TYPE="SUBMIT" VALUE="Get the weather">
</FORM>
```

A browser receiving this form, or Bob if he views the source of the form, is assured that the assigning authority is supporting an entire class of URIs of the form:

```
http://example.org/cityweather?city=CityName
```

The same HTML Form is also a computer program, executable by the browser, that prompts for and retrieves representations for all such URIs, and the English text in the form assures Bob that these are indeed for weather reports. Bob is not guessing the encoding of the URI or the nature of the resources referenced — he is acting on authoritative information provided by the assigner of the URIs. He can assume not just that he will get weather reports for certain cities, but that no URIs in the class correspond to anything other than weather reports (though some may correspond to no resource at

all). Bob could, with this assurance, write his own software to construct and use such URIs to retrieve weather reports. Of course, the typical Web user would neither directly inspect the URIs nor write software to build them, but would instead type in city names and push the handy "Get the weather" button on his or her browser screen.

Comment [skw9]: Ok… but Bob's software is also vulnerable to change *if* example.org change the way that they organise their URI space (modulo or not "Cool URIs…"). I think that this risks overstating the assurance that Bob has.

Note that the example carefully specifies that the HTML form is sourced from the same authority as the individual weather URIs that the form queries. In fact, it is also common for the `ACTION` attributes in HTML forms to refer to URIs from other authorities. In such cases, it is the provider of the form rather than the assigning authority for the queried URIs that is responsible for the claims made in the form. In particular, users (and software) should check the origin of HTML forms before depending on the URI assignment patterns that they appear to imply. Of course, you can always use such a form to perform a query and see what comes back; what you can't do is blame the assignment authority if the generated URIs either don't resolve (status code 404) or return representations that don't match the expectations established when reading the form (you got a football score instead of a weather report).

2.5 Authority use of URI metadata

In the examples above, resource metadata (I.e. the city associated with each resource) was encoded into URIs primarily for the benefit of users such as Bob, or to facilitate use of the HTML Forms or XForms acting on those users' behalf.

Often, metadata is encoded into a URI not primarily for the benefit of users, but to facilitate management of the resources themselves. For example, assume that the administrators at example.org have established a policy of assigning URIs based on the media types of representations: all GIF images are named with URIs ending in ".gif", and all JPEG images are named with URIs ending in ".jpeg", and so on. Although **2.1 Reliability of URI metadata** warned that *users* of a resource cannot rely on undocumented naming conventions to determine media types and other information about a resource, the *owner* of a resource controls such naming and can depend on it. Example.org may therefore rely on their policy in an Apache Web Server .htaccess file, which causes the correct media type to be served automatically for each resource:

```
<Files ~ ".*\.gif">
  ForceType 'image/gif'
</Files>
<Files ~ ".*\.jpg">
  ForceType 'image/jpeg'
</Files>
```

Even if it does not document this policy publicly, example.org's own Web servers can safely depend on it.

Good Practice

Good Practice: URI assignment authorities and the Web servers deployed for them may benefit from an orderly mapping from resource metadata into URIs.

In addition to filename-based conventions, authorities may choose to base URIs on database keys, customer identifiers, or other information that makes it easy to associate a URI with information pertinent to the corresponding resource. Such encodings are both useful and common on the Web, but there can also be drawbacks to including such information in URIs. Some of those problems are discussed in the three sections immediately below.

2.6 URIs that are convenient for people to use

URIs optimized for use by the assignment authority may sometimes be inconvenient for resource users. Consider Mary who is walking down the street, and who sees the same weather advertisement as Bob:

"For the best Chicago Weather information on the Web, visit http://example.org/weather/Chicago."

Like Bob, Mary is pleased to learn about a valuable Web site, and she finds that the URI itself is quite easy to both to remember and to type into her browser. This is because, in addition to the required scheme and authority components, the URI is based on the word *weather* and the city name *Chicago*, both of which fit her expectations for this resource.

The next day, Mary sees another advertisement reading:

"For the best Atlanta Weather information on the Web, visit http://example.org/123Hx67v4gZ5234Bq5rZ."

Mary is annoyed, because the URI is both difficult to remember and hard to transcribe accurately. She guesses that the authority has assigned this URI for its own convenience (see **2.5 Authority use of URI metadata**) rather than for hers. Although Web architecture does not require that URIs be easy to understand or suggestive of the resource named, it's handy if those intended for direct use by people are.

Good Practice

Good Practice: URIs intended for direct use by people should be easy to understand, and should be suggestive of the resource actually named.

Note that the second URI might be based on a database key that facilitates efficient access to the weather data at the server (see **2.5 Authority use of URI metadata**); such a URI might have been a good choice if it were intended only for use in HTML hyperlinks, rather than in an advertisement on the side of a bus.

2.7 Changing metadata

URIs should generally not encode metadata that will change, regardless of whether the encoding policy is established to benefit URI assignment authorities, resource users, or both. Consider a web site that organizes document URIs according to the documents' lead author or editor. Thus, the documents:

```
http://example.org/documents/editor/BobSmith/document1
http://example.org/documents/editor/BobSmith/document2
```

are named for their editor, Bob Smith. Bob retires, and Mary Jones takes over as editor for document1. If the URI is changed to encode her name, then existing links break, but if the URI is not changed, the naming policy is violated. By encoding into the URI metadata that will change, the authority has put itself in a difficult position.

Good Practice

Good Practice: Resource metadata that will change SHOULD NOT be encoded in a URI.

Indeed, RDF statements about the resource, headers returned with representations (e.g. Content-Type) or metadata embedded in the representations themselves (e.g. HTML <META> tags) are all better alternatives for conveying such volatile metadata about the resource.

2.8 Hiding metadata for security reasons

A bank establishes a URI assignment policy in which account numbers are encoded directly in the URI. For example, the URI http://example.org/customeraccounts/456123 accesses information for account number 456123. A malicious worker at an Internet Service Provider notices these URIs in his traffic logs, and determines the bank account numbers for his Internet customers. Furthermore, if access controls are not properly in place, he might be able to guess the URIs for other accounts, and to attempt to access them.

Good Practice

Good Practice: URI assignment authorities should not put into URIs metadata that is to be kept confidential.

3 Conclusions

The principle conclusions of this finding are:

- It is legitimate for assignment authorities to encode static identifying properties of a resource, e.g. author, version, or creation date, within the URIs they assign. This may contribute to the unique assignment of URIs. It may also contribute to the use of efficient mechanisms for dereferencing resources within origin servers e.g. use of database keys within URIs.
- Assignment authorities may publish specifications detailing the structure and semantics of the URIs they assign. Other users of those URIs may use such specifications to infer information about resources identified by URI assigned by that authority.
- The ability to explore and experiment is important to Web users. Users therefore benefit from the ability to infer either the nature of the named resource, or the likely identity of other resources, from inspection of a URI. Such inferences are reliable only when supported by normative specifications or by documentation from the assignment authorities. In other cases, users are responsible for the consequences of any incorrect inferences.
- People and software using URIs assigned outside of their own authority should make as few inferences as possible about a resource based on its identity. The more dependencies a piece of software has on particular constraints and inferences, the more fragile it becomes to change and the lower its generic utility.

Formatted: Bullets and Numbering

Comment [skw10]: I think that the generation of unique identifiers is the more likely reason for embedding so-called metadata in a URI. I suspect that in general it is rarely the intent that the URI be parsed to extract what some construe as embedded 'metadata'.

I think the uniqueness driver should be introduced earlier, where sufficient static distinguishing characteristics are encoded into a URI in order to make it unique.

Comment [skw11]: I think that given that such specifications may be subject to change, there should be some caution suggested wrt the permanence of any implied commitment on the part of the assignment authority.

4 References

AUTHMETA
"Authoritative Metadata"; W3C; TAG Finding; R.T. Fielding, I.Jacobs; April 2006 (See http://www.w3.org/2001/tag/doc/mime-respect.)

HTTP
"Hypertext Transfer Protocol - HTTP/1.1"; IETF; RFC 2616; R. Fielding, J. Gettys, J. Mogul, H. Frystyk, P. Leach, L. Masinter, T. Berners-Lee; June 1999 (See http://www.iana.org/rfc/rfc2616.)

HTMLFORMS
"HTML 4.01 Specification (Forms Chapter)"; W3C; D. Raggett, A. Le Hors, I. Jacobs; December 1999 (See http://www.w3.org/TR/html4/interact/forms.html.)

RFC2119
S. Bradner. Key words for use in RFCs to Indicate Requirement Levels. IETF. March, 1997. (See http://www.ietf.org/rfc/rfc2119.txt.)

URI
"Uniform Resource Identifiers (URI): Generic Syntax"; RFC3986; IETF; T. Berners-Lee, R. Fielding, L. Masinter; August 1998 (See http://www.ietf.org/rfc/rfc3986.)

XFORMS
"XForms 1.0"; W3C; J.M. Boyer, D. Landwehr, R. Merrick, T. V. Raman, M. Dubinko, L. Klotz ; 2006 (2nd Edition) (See http://www.w3.org/TR/xforms/.)