LIS 900C Webmastering I: the static web page

Lecture 2

URI

Thomas Krichel

2002–05–13

Reading

Berners-Lee, Tim Roy T. Fielding and Larry Masinter (1998)
"Uniform Resource Identifiers (URI): Generic Syntax", rfc2396

Structure

1.
2.
3.

Definition

A Uniform Resource Identifier (URI) is a compact string of characters for identifying an abstract or physical resource. They provide a simple and extensible means for identifying a resource.

There is a set of operations that can be applied to them.

To understand if a given URI instance is valid, we have to study the operations applied to URIs.

uniformity

Uniformity provides several benefits:

- it allows different types of resource identifiers to be used in the same context, even when the mechanisms used to access those resources may differ
- it allows uniform semantic interpretation of common syntactic conventions across different types of resource identifiers
- allows introduction of new types of resource identifiers without interfering with the way that existing identifiers are used;
- it allows the identifiers to be reused in many different contexts, thus permitting new applications or protocols to leverage a pre-existing, large, and widely-used set of resource identifiers.

resource identifer

A resource can be anything that has identity. Not all resources are network "retrievable". The resource is the conceptual mapping to an entity or set of entities, not necessarily the entity which corresponds to that mapping at any particular instance in time.

An identifier is an object that can act as a reference to something that has identity. In the case of URI, the object is a sequence of characters with a restricted syntax.

URI, URL, URN

A URI can be further classified as a locator, a name, or both. The term "Uniform Resource Locator" (URL) refers to the subset of URI that identify resources via a representation of their primary access mechanism (e.g., their network "location"), rather than identifying the resource by name or by some other attribute(s) of that resource. The term "Uniform Resource Name" (URN) refers to the subset of URI that are required to remain globally unique and persistent even when the resource ceases to exist or becomes unavailable.

URN

A URN differs from a URL in that it's primary purpose is persistent labeling of a resource with an identifier. That identifier is drawn from one of a set of defined namespaces, each of which has its own set name structure and assignment procedures. The "urn" scheme has been reserved to establish the requirements for a standardized URN namespace, as defined in "URN Syntax" RFC2141 and its related specifications.

absolute and relative identifiers

An absolute identifier refers to a resource independent of the context in which the identifier is used. In contrast, a relative identifier refers to a resource by describing the difference within a hierarchical namespace between the current context and an absolute identifier of the resource.

Some URI schemes support a hierarchical naming system, where the hierarchy of the name is denoted by a "/" delimiter separating the components in the scheme.

transcribability

The URI syntax was designed with global transcribability as one of its main concerns. A URI is a sequence of characters from a very limited set, i.e. the letters of the basic Latin alphabet, digits, and a few special characters. A URI may be represented in a variety of ways. Therefore:

- A URI is a sequence of characters, which is not always represented as a sequence of octets.
- A URI may be transcribed from a non-network source, and thus should consist of characters that are most likely to be able to be typed into a computer, within the constraints imposed by keyboards (and related input devices) across languages and locales.
- A URI often needs to be remembered by people, and it is easier for people to remember a URI when it consists of meaningful components.

URI characters

URI consist of a restricted set of characters, not a sequence of octets. The allowable characters primarily chosen to aid transcribability and usability both in computer systems and in non-computer communications. Characters used conventionally as delimiters around URI are excluded.

In the simplest case, the original character sequence contains only characters that are defined in US-ASCII, and the two levels of mapping are simple and easily invertible: each 'original character' is represented as the octet for the US-ASCII code for it, which is, in turn, represented as either the US-ASCII character.

reserved characters

Many URI include components consisting of or delimited by, certain special characters. These characters are called "reserved", since their usage within the URI component is limited to their reserved purpose. If the data for a URI component would conflict with the reserved purpose, then the conflicting data must be escaped before forming the URI.

; / ? : @ & = + $,

The "reserved" syntax class above refers to those characters that are allowed within a URI, but which may not be allowed within a particular component of the generic URI syntax.

unreserved characters

URI consist of a restricted set of characters, not a sequence of octets. The allowable characters primarily chosen to aid transcribability and usability both in computer systems and in non-computer communications. Characters used conventionally as delimiters around URI are excluded.

In the simplest case, the original character sequence contains only characters that are defined in US-ASCII, and the two levels of mapping are simple and easily invertible: each 'original character' is represented as the octet for the US-ASCII code for it, which is, in turn, represented as either the US-ASCII character.

Unreseved and excluded characters

Unreserved characters are allowed and do not have special meaning. The are the upper and lower case letters, decimal digits, the following

- _ . ! ~ * ' ( )

All other characters are excluded. In particular

< > # % " { } | ^ [] ‘ and the blank

are excluded. They have to be escaped.

Escape sequence

When you want to use a character in a URI that not one of the unreserved characters. The way that this done is to write a construction of the form

%*hex hex*

where *hex* is a digit or the letters a to f (uppercase or lowercase). The two hex characters represent the value of the character in hex. For example **%7e** is the character ~.

URI components.

In general, an absolute URI is written as follows

scheme:*scheme_specific_part*

However, an important subset of URIs, have more structure. This subset is what we call a the generic syntax

scheme://*authority path* ? *query*

All component except *scheme* are optional.

A slash / is used to express hierarchies.

scheme component.

A scheme is a way to identify a resource. Scheme names must start with a lower case letter, followed by other letters, digits and +,−, or .

Relative URI do not start with a scheme name.

authority component

It can either be a name, or—when it refers to location on the Internet—take the form

[*userinfo*@]*host*[:*port*]

where components in the square brackes can be ommitted. *host* is an Internet host either identified through its DNS name or through its IP address, as described in described in Section 3 of RFC1034 and Section 2.1 of RFC1123, respectively.

path component

The path component contains data, specific to the authority (or the scheme if there is no authority component), identifying the resource within the scope of that scheme and authority.

The path may consist of a sequence of path segments separated by a single slash / character. Within a path segment, the characters / , ;, =, and ? are reserved.

query component

The query component is a string of information to be interpreted by the resource. It is often of the form

action?*parameters*

where *parameters* are of the form

name=*value*&

where *name* is a parameter name and *value* is the value that the parameter takes.

URI references

It is common that there is some action to be performed after the retrieval of the object is performed. In that case, an indicator of the action to be performed is appended at the end of the URI, separated by the **#** character.

A URI reference that does not contain a URI is a reference to the current document. Such references occur frequently within html documents.

Relative URIs

It is often the case that a group or "tree" of documents has been constructed to serve a common purpose; the vast majority of URI in these documents point to resources within the tree rather than outside of it. In this case relative URIs are used. That will allow the doucemnts to be moved easily.

Within html documents that are static within file trees, **/** refers to the server root, directories are separated by slashes, **..** is the expression to go to a higher level directory, and **.** is the current directory.