Uniform Resource Identifier (URI)

Péter Jeszenszky
Faculty of Informatics, University of Debrecen
jeszenszky.peter@inf.unideb.hu

Last modified: September 13, 2024

URI (1)

- Uniform Resource Identifier (URI):
  - A compact sequence of characters that identifies an abstract or physical resource.
    - A resource is not necessarily available on the Web.
    - URIs can be assigned even to objects from the real world or to concepts.
- Current standard:
  - Tim Berners-Lee, Roy Fielding, Larry Masinter. *RFC 3986: Uniform Resource Identifier (URI): Generic Syntax*. January 2005. https://www.rfc-editor.org/rfc/rfc3986

URI (2)

- Each URI begins with a scheme name that is separated by a `':'` character from the scheme-specific part of the URI.
  - Scheme specifications can define their scheme-specific syntax within certain limits.
- The organization responsible for the administration of the URI schemes:
  - Internet Assigned Numbers Authority (IANA) https://www.iana.org/
    - See: *Uniform Resource Identifier (URI) Schemes* https://www.iana.org/assignments/uri-schemes/uri-schemes.xhtml

Well-Known URI Schemes

- `file:`
  - Matthew Kerwin. *RFC 8089: The "file" URI Scheme*. February 2017. https://www.rfc-editor.org/rfc/rfc8089
- `http`/`https`:
  - Roy T. Fielding (ed.), Mark Nottingham (ed.), Julian F. Reschke (ed.). *RFC 9110: HTTP Semantics*. June 2022. https://www.rfc-editor.org/rfc/rfc9110
- `mailto`:
  - Martin Dürst, Larry Masinter, Jamie Zawinski. *RFC 6068: The 'mailto' URI Scheme*. October 2010. https://www.rfc-editor.org/rfc/rfc6068
- `about`:
  - S. Moonesamy (ed.). *RFC 6694: The “about” URI Scheme*. August 2012. https://www.rfc-editor.org/rfc/rfc6694

Dereferencing

- Accessing the resource identified by a URI.
  - In most cases, “access” means the retrieval of a representation of the resource.

URL vs URN (1)

- Historically, two disjoint types of URIs are distinguished:
  - ***Uniform Resource Locator*** (**URL**):
    - Identifying resources by their location.
    - Tim Berners-Lee, Larry Masinter, Mark P. McCahill. *RFC 1738: Uniform Resource Locators (URL)*. December 1994. https://www.rfc-editor.org/rfc/rfc1738
  - ***Uniform Resource Name*** (**URN**):
    - Persistent and location-independent resource identifiers.
    - Ryan Moats. *RFC 2141: URN Syntax*. May 1997. https://www.rfc-editor.org/rfc/rfc2141

URL vs URN (2)

- This former classification is now obsolete:
  - Michael Mealling (ed.), Ray Denenberg (ed.). *RFC 3305: Report from the Joint W3C/IETF URI Planning Interest Group: Uniform Resource Identifiers (URIs), URLs, and Uniform Resource Names (URNs): Clarifications and Recommendations*. August 2002. https://www.rfc-editor.org/rfc/rfc3305
  - *URIs, URLs, and URNs: Clarifications and Recommendations 1.0—Report from the joint W3C/IETF URI Planning Interest Group* (W3C Note, 21 September 2001) https://www.w3.org/TR/uri-clarification/

URL vs URN (3)

- According to the contemporary view, a URI can be a locator, a name, or both at the same time.
  - A URI scheme does not need to be cast into one of a discrete set of URI types, such as URL or URN.
- URL is an informal concept, it means a URI that identifies a resource via a representation of its primary access mechanism (for example, its network “location”).

URN

- A Uniform Resource Name (URN) is a persistent, location-independent resource identifier.
- A URN is a URI that is assigned under the `urn` URI scheme.
- See:
  - Peter Saint-Andre, John C. Klensin. *RFC 8141: Uniform Resource Names (URNs)*. April. 2017. https://www.rfc-editor.org/rfc/rfc8141

WHATWG standard (1)

- *URL Living Standard*
  https://url.spec.whatwg.org/
- Goals:
  - Align RFC 3986 and RFC 3987 with contemporary implementations and obsolete them in the process.
  - Standardize on the term URL.
  - Enhance URL's existing JavaScript API.

WHATWG standard (2)

- Handle URIs and IRIs uninformly.
- A URL is a universal identifier.

URI vs URL (IETF vs WHATWG)

- See:
  - Daniel Stenberg. *My URL isn't your URL*. May 11, 2016. https://daniel.haxx.se/blog/2016/05/11/my-url-isnt-your-url/
  - Daniel Stenberg. *One URL standard please*. January 30, 2017. https://daniel.haxx.se/blog/2017/01/30/one-url-standard-please/
  - Daniel Stenberg. *URL Interop*. https://github.com/bagder/docs/blob/master/URL-interop.md

URI Examples

- https://www.rfc-editor.org/rfc/rfc3986.txt
- https://url.spec.whatwg.org/#references
- file:///usr/lib/R/library
- about:downloads
- mailto:jeszenszky.peter@inf.unideb.hu
- ldap://ldap.iplanet.com/dc=example,dc=com
- tel:+36-52-512-900
- news:comp.lang.c
- urn:isbn:0-395-36341-1
- urn:ietf:std:66
- urn:uuid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6
- geo:47.5539464,21.6215658

URI Characters (1)

- Characters allowed in URIs:
  - The following are reserved characters:
    - `':', '/', '?', '#', '[', ']', '@', '!', '$', '&', ''', '(', ')', '*', '+', ',', ';', '='`
      - Characters used as delimiters.
  - The following are unreserved characters:
    - `'A', …, 'Z', 'a', …, 'z'`
    - `'0', …, '9'`
    - `'-', '.', '_', '~'`
- Thd specification does not mandate any particular character encoding.

URI Characters (2)

- **Percent-encoding**: used to represent a data octet in a component when that octet's corresponding character is outside the allowed set or is being used as a delimiter of, or within, the component.
  - A percent-encoded octet is encoded as a character triplet *%hh*, consisting of the `'%'` character followed by the two hexadecimal digits representing that octet's numeric value.
    - For example, `%20` is the percent-encoding the space character.
    - Both the uppercase (`'A'`, …, `'F'`) and the lowercase (`'a'`, …, `'f'`) hexadecimal digits can be used.
    - If two URIs differ only in the case of hexadecimal digits used in percent-encoded octets, they are equivalent.

URI Characters (3)

- Percent-encoding examples:
  - `file:///media/Movies/What's Up, Tiger Lily? (1966)/` → `file:///media/Movies/What%27s%20Up%2C%20Tiger%20Lily%3F%20%281966%29/`
  - Assuming UTF-8 character encoding: `http://www.w3.org/People/Dürst/` → `http://www.w3.org/People/D%C3%BCrst/`

URI Syntax (1)

- Syntax is organized hierarchically.
  - Components listed in order of decreasing significance from left to right.
- Generic syntax:

  scheme `':'` *hier-part* [`'?'` *query*] [`'#'` *fragment*]

  - The hier-part component may consist of an authority and a path component, its syntax is:

    `'//'` *authority path* or *path*

    - When authority is present, the path must either be empty or begin with a `'/'` character.
    - When authority is not present, the path cannot begin with two `'/'` characters.

URI Syntax (2)

- Example:

```
https://wordery.com/search?term=scotland#results
\___/   \_____________/\_____/ \_____________/ \_____/
  |            |         |           |           |
scheme       host      path        query     fragment
```

- Example:

```
mailto:jeszenszky.peter@inf.unideb.hu?subject=URI
\____/ \_____________________________________/ \_____/
   |                      |                       |
scheme                  path                    query
```

URI Syntax (3)

- Fun fact:
  - In an article published in the Times in October 2009, Tim Berners-Lee apologized for the two slashes in URI-s:
    - “There you go, it seemed like a good idea at the time…”
  - Quoted in: *Berners-Lee 'sorry' for slashes*. 14 October 2009. http://news.bbc.co.uk/2/hi/technology/8306631.stm

Authority

- The name comes from that the name space defined by the remainder of the URI is under its jurisdiction.
- Syntax:
  [*userinfo* `'@'`] *host* [`':'` *port*]
  - A URI scheme may define a default port.
    - For example, the `http` scheme defines a default port of 80.

Path

- A sequence of path segments separated by a `'/'` character.
- Terminated by the first `'?'` or `'#'`, or by the end of the URI.
- The path segments `'.'` and `'..'` can be used just as in some operating systems' file directory structures.

Query

- Indicated by the first `'?'` character and terminated by a `'#'` character or by the end of the URI.
- Contains non-hierarchical data.
- Often contains name/value pairs of the form *name* `'='` *value* delimited by an `'&'` character.
  - In the case of the `http` and `https` URI schemes the query component is used for submitting form data (see the `application/x-www-form-urlencoded` format).
  - Example:
    - https://blackwells.co.uk/bookshop/search?keyword=sherlock+holmes&sortValue=DateDesc
  - See: *HTML Standard – URL-encoded form data* https://html.spec.whatwg.org/multipage/forms.html#url-encoded-form-data

Fragment Identifier (1)

- Indicated by a `'#'` character and terminated by the end of the URI.
- Allows indirect identification of a secondary resource by reference to a primary resource and additional identifying information.
  - The identified secondary resource may be some portion or subset of the primary resource, some view on representations of the primary resource, or some other resource defined or described by those representations.
- The semantics of a fragment identifier are defined by the set of representations that might result from a retrieval action on the primary resource.
  - Media types may also define their own restrictions on or structures within the fragment identifier syntax.
- The fragment identifier is separated from the rest of the URI prior to a dereference.

Fragment Identifier (2)

- URI scheme specifications must define their own syntax so that all strings matching their scheme-specific syntax must be an absolute URI without a fragment identifier.
  - Scheme specifications will not define fragment identifier syntax or usage, regardless of its applicability to resources identifiable via that scheme, as fragment identification is orthogonal to scheme definition.

Meaning of the Fragment Identifier (1)

- `text/html` media type:
  - Fragment identifiers either refer to the indicated part of the document or provide state information for in-page scripts. https://www.iana.org/assignments/media-types/text/html
  - Detailed processing for fragment identifiers is defined in the HTML5 specification.
    - See: *Navigating to a fragment* https://html.spec.whatwg.org/multipage/browsing-the-web.html#scroll-to-fragid
  - For example, the fragment identifier in the https://www.mozilla.org/en-US/#colophon URI refers to the element with `id="colophon"`.
  - For example, the fragment identifier in the https://www.youtube.com/watch?v=w0ffwDYo00Q#t=77 URI indicates the position from which playback will be started (at the 77th second).

Meaning of the Fragment Identifier (2)

- `application/xml`, `text/xml` and `*/*+xml` media types:
  - The latter includes, for example, the following media types: `application/xhtml+xml`, `image/svg+xml`, `model/x3d+xml`
  - The syntax and semantics of fragment identifiers is based on the XPointer Framework specification. https://www.iana.org/assignments/media-types/text/xml
    - *XPointer Framework* (W3C Recommendation, 25 March 2003) https://www.w3.org/TR/xptr-framework/
  - For example, the fragment identifier in the https://www.w3.org/TR/xml/#sec-bibliography URI refers to the element with identifier `sec-bibliography` in the document.

Absolute URI, URI-reference, relative reference

- **Absolute URI**: a URI without a fragment identifier.
  - Only absolute URIs can be used as a base URI.
- **URI-reference**: a URI or a relative reference.
- **Relative reference**: a scheme-specific subpart of a URI or a suffix of it (can be empty).
  - The specification does not use the term “relative URI” at all!
  - URIs are interpreted consistently regardless of context, relative references are interpreted in a context.
  - Relative references are resolved to a URI against a base URI. The resulting URI is also known as the target URI.
  - The specification describes an algorithm for resolving relative references.

URI-reference Examples

- `http://www.gnu.org/licenses/licenses.html`
- `http://www.w3.org/TR/xml/#abstract`
- `http://en.wikipedia.org/wiki/The_Beatles#History`
- `/pub/linux/kernel/v3.x/testing/`
- `../../images/bullet.png`
- `index.html#contents`
- `contacts.xml#element(/1/2)`
- `#nav`
- `gpl.html`
- ⟨*empty string*⟩

Same-document reference

- A URI-reference that refers to a URI that is, aside from its fragment component (if any), identical to the base URI.
  - Example: ⟨*empty string*⟩, #nav
  - A dereference should not result in a new retrieval action.

Establishing a Base URI

- Within certain media types, a base URI for relative references can be embedded within the content itself.
  - Thus, for example, documents can define their base URI.
    - XML: the base URI can be specified by the `xml:base` attribute (see later).
    - HTML: the base URI can be provided by the `base` element. https://html.spec.whatwg.org/multipage/semantics.html#the-base-element
- If no base URI is embedded and a representation is enclosed within another entity – for example, another document –, then the base URI is the base URI of the entity in which the representation is encapsulated.
- If no base URI is embedded and the representation is not encapsulated within some other entity, then, if a URI was used to retrieve the representation, that URI is considered the base URI.
  - If the retrieval was the result of a redirected request, the last URI used is the base URI.
- Otherwise, the base URI is application-dependent.

Relative Reference Resolution Examples (1)

- Let `http://example/a/b/c?q` be the base URI

Relative Reference	Target URI
`d`	`http://example/a/b/d`
`./d`	`http://example/a/b/d`
`/d`	`http://example/d`
`//localhost`	`http://localhost`
`?y`	`http://example/a/b/c?y`
`d?y`	`http://example/a/b/d?y`

Relative Reference Resolution Examples (2)

- Let `http://example/a/b/c?q` be the base URI

Relative Reference	Target URI
`#z`	`http://example/a/b/c?q#z`
`""` (empty string)	`http://example/a/b/c?q`
`.`	`http://example/a/b/`
`./`	`http://example/a/b/`
`..`	`http://example/a/`
`../d`	`http://example/a/d`
`../../d`	`http://example/d`

Relative Reference Resolution Examples (3)

- Example:
  - ```
    <!DOCTYPE html>
    <html lang="en">
      <head>
        <title>Example</title>
        <base href="http://example/docs/howto/">
        <link rel="stylesheet" type="text/css" href="theme.css">
      </head>
      <body>
        <a href="/about">
          <img src="../images/logo.png" alt="Logo">
        </a>
      </body>
    </html>
    ```
  - Resolution of the relative references:
    - `theme.css` → `http://example/docs/howto/theme.css`
    - `/about` → `http://example/about`
    - `../images/logo.png` → `http://example/docs/images/logo.png`
    ```

URI Comparison (1)

- The scheme and host components are case-insensitive.
- The other syntax components are assumed to be case-sensitive unless specifically defined otherwise by the scheme.
- For example, the http://www.w3.org/ and HTTP://www.W3.org/ URIs are equivalent.

URI Comparison (2)

- A possible definition of equivalence:
  - URIs should be considered equivalent when they identify the same resource.
  - This definition is not of much practical use, because in general there is no way to compare two resources.
- In practice, equivalence is determined by string comparison.
  - Normalization is applied before comparison, for example, uppercase letters are converted to lowercase letters in case-insensitive components.

JavaScript API

- The URL Living Standard of WHATWG defines an API to work with and manipulate URLs in JavaScript: https://url.spec.whatwg.org/#api
- See also: https://developer.mozilla.org/en-US/docs/Web/API/URL
- Example:

```
const url = new URL(
  "../images",
  "https://eg.com/docs/index.html"
);
console.log(url.href);
```

Internationalized Resource Identifier (1)

- IRIs consist of Unicode/UCS characters instead of US-ASCII characters.
- The syntax and use of components and reserved characters is the same as that in the URI specification.
- The range of unreserved characters is expanded to include Unicode/UCS characters.
- Current standard:
  - Martin Dürst, Michel Suignard. *RFC 3987: Internationalized Resource Identifiers (IRIs)*. January 2005. https://www.rfc-editor.org/rfc/rfc3987

Internationalized Resource Identifier (2)

- Examples:
  - https://en.wiktionary.org/wiki/γεια
  - http://www.öbb.at/

Internationalized Resource Identifier (3)

- An IRI reference can be mapped to an equivalent URI reference:
  - For each character of the IRI reference that is not allowed in URI references perform the following steps:
    - Convert the character to a sequence of one or more octets using UTF-8.
    - Convert each octet to %*HH*, where *HH* is the hexadecimal notation of the octet value (percent-encoding).
    - Replace the original character with the resulting character sequence.
  - Example: https://www.w3.org/People/Dürst/ → https://www.w3.org/People/D%C3%BCrst/

Internationalized Resource Identifier (4)

- Advantages:
  - Ease the use of URIs for users whose language uses an alphabet other than Latin.
- Risks:
  - Homograph attacks: tricking users by exploiting the fact that there are different Unicode characters that look alike.
    - The risk also exists with URIs, see, for example, `lame` and 1ame, broken and br0ken.

XML Base (1)

- *XML Base (Second Edition)* (W3C Recommendation, 28 January 2009) https://www.w3.org/TR/xmlbase/
  - Provides a mechanism for defining base URIs in XML documents.
  - Introduces the `xml:base` attribute to specify a base URI.
    - The attribute is inherited by descendant elements until another element with an `xml:base` attribute is encountered.

XML Base (2)

- Example:

```
<book xml:base="http://example/books/untitled/"
  xmlns:xi="http://www.w3.org/2001/XInclude">
  <xi:include href="info.xml"/>
  <xi:include href="chapter1.xml"/>
  <xi:include href="../thanks.xml"/>
  <bibliography xml:base="/biblio/">
    <xi:include href="unsorted.xml"/>
  </bibliography>
</book>
```

  - Resolution of the relative references:
    - `info.xml` → `http://example/books/untitled/info.xml`, `chapter1.xml` → `http://example/books/untitled/chapter1.xml`
    - `../thanks.xml` → `http://example/books/thanks.xml`
    - `unsorted.xml` → `http://example/biblio/unsorted.xml`

URL Shortening (1)

- Long `http(s)` URIs can be shortened using HTTP redirection.
- The aim of URL shortening is to create an `http(s)` URI that points to the same resource, but is more aesthetic, more compact and can be displayed and communicated more easily.
  - Originally and historically, the length of Twitter messages was limited to 140 characters.
    - See: *Giving you more characters to express yourself* https://blog.twitter.com/official/en_us/topics/product/2017/Giving-you-more-characters-to-express-yourself.html

URL Shortening (2)

- A list of URL shorteners: https://github.com/738/awesome-url-shortener
- A number of websites provides URL shortening functionality:
  - GitHub: https://git.io/
  - Twitter: https://t.co/
  - Wikipedia: https://meta.wikimedia.org/wiki/Special:UrlShortener
  - …

URL Shortening: TinyURL

Owner:	TinyURL, LLC.
Web page:	https://tinyurl.com/
HTTP status code:	`301` (`Moved Permanently`)
Registration:	no
Custom URI:	yes
URL information:	yes
Tracking:	no
API:	https://tinyurl.com/app/dev
Examples:	https://tinyurl.com/y3rznxva https://preview.tinyurl.com/y3rznxva https://tinyurl.com/IntStream https://preview.tinyurl.com/IntStream

URL Shortening: Bitly

Owner:	Bitly, Inc.
Web page:	https://bitly.com/
HTTP status code:	`301 (Moved Permanently)`
Registration:	yes
Custom URI:	yes
URL information:	yes
Tracking:	yes
API:	https://dev.bitly.com/
Examples:	https://bit.ly/2FlomT4 https://bit.ly/2FlomT4+ https://amzn.to/3h8qX00 https://amzn.to/3h8qX00+