References Names Pointers Paths URI URN URL CURIE QName

URL [RFC 3986]

Uniform Resource Locator, part of REST

scheme://authority/path?query#fragment-id

Can be dereferenced:

- Talk to authority using protocol implied by scheme
- select resource via path, query
- fragment-id is **local** pointer (stepchild of URI)

URI reference [RFC 3986]

Generate a URI from a base and a reference
Can leave off parts that will be filled from context

There can only be one base for a context
(Usually the document in which reference is embedded)

URN [RFC 3986, urn: RFC 8141]

Uniform Resource Name, part of REST

Using URI syntax without intention of dereferencing
Does not need authority

Confusion: URN as a concept vs. urn: scheme

Fragment identifier weirdness

Fragment identifiers are interpreted in the media type of the referenced representation
➔ Does not mean anything if there is no dereferencing

Still, widely used with URNs (e.g., RDF 1.0:)

a URI reference in an RDF graph is treated with respect to the MIME type application/rdf+xml

(RDF 1.1 acknowledges that RDF notes can be URIs to non-RDF documents!)

IRIs, IRI references [RFC 3987]

URIs are ASCII
can encode UTF-8 (not guaranteed): percent-encoding

IRIs are like URIs, but based on Unicode
(UTF-8 in interchange)

URLs in practice [WHATWG] are IRIs

XML namespaces

QName: URI + local Name
Namespace name: URI (really: URN!)
Namespace prefix: short name of URI for QName
compression
xmlns:svg="http://www.w3.org/2000/svg" ➔ svg:ellipse
Syntax without compression: `{http://www.w3.org/2000/svg}ellipse`

XML allows QNames in arbitrary strings:
Application needs access to namespace declarations

RDF

RDF uses URIs as URNs, usually with fragment ids
`rdf:type` → RDF URI reference (encoded as URI ~ IRI)
`http://www.w3.org/1999/02/22-rdf-syntax-ns#type`

Concatenate namespace name and local name into RDF URI reference by simple concatenation, creating boundary weirdness (RDF 1.0):

»Within RDF/XML documents it is not permitted to use XML namespaces whose namespace name is the ·RDF namespace URI reference· concatenated with additional characters.«

CURIEs

Make RDF-style URI referencess available outside RDF:
CURIE: [[prefix] :] IRI-Reference

Not the same as QNames!
but an XML vocabulary can still use QNames for CURIEs

Unless "safe" CURIEs (with brackets around) are used:
CURIEs cannot be mixed with absolute URLs

Decompress by concatenating name for prefix and IRI-Reference
Compress by ??? (insert human here to choose prefixes)

Prefix name weirdness

Many interpreters rely on specific prefixes for XML, RDF, CURIE

Mixing data from different sources requires decompression/ compression

https://www.drupal.org/docs/7/api/rdf-mapping-api/rdf-namespaces-registry

TL;DR: If your module needs to use one of the prefixes below, then that prefix should be mapped to the 'canonical' namespace listed below. This will help prevent conflicts with other modules that might map the same prefix to a different namespace.

```
/**
 * Implements hook_rdf_namespaces().
 */
function rdf_rdf_namespaces() {
  return array(
    // Core RDF namespaces which don't need to be redefined in modules.
    'content'  => 'http://purl.org/rss/1.0/modules/content/',
    'dc'       => 'http://purl.org/dc/terms/',
    'foaf'     => 'http://xmlns.com/foaf/0.1/',
    'og'       => 'http://ogp.me/ns#',
    'rdfs'     => 'http://www.w3.org/2000/01/rdf-schema#',
    'sioc'     => 'http://rdfs.org/sioc/ns#',
    'sioct'    => 'http://rdfs.org/sioc/types#',
    'skos'     => 'http://www.w3.org/2004/02/skos/core#',
    'xsd'      => 'http://www.w3.org/2001/XMLSchema#',
    // RDF namespaces that contributed and custom modules should use if needed.
    // Only define the ones that are actually use in your module.
    'schema'   => 'http://schema.org/',
    'rnews'    => 'http://iptc.org/std/rNews/2011-10-07#',
    'dbp'      => 'http://dbpedia.org/property/',
    'grddl'    => 'http://www.w3.org/2003/g/data-view#',
    'ma'       => 'http://www.w3.org/ns/ma-ont#',advanced_help
    'owl'      => 'http://www.w3.org/2002/07/owl#',
    'prov'     => 'http://www.w3.org/ns/prov#',
    'rdf'      => 'http://www.w3.org/1999/02/22-rdf-syntax-ns#',
    'rdfa'     => 'http://www.w3.org/ns/rdfa#',
    'rif'      => 'http://www.w3.org/2007/rif#',
```

Computing URIs: URI templates [RFC 6750]

RFC 7320 (BCP 190): Servers provide URIs into them

URIs can be used as containers for protocol parameters
(RFC 6750: arcane syntax, multiple "levels")
"Macro language", some processing implied (level 4!)
Client-side computation

Variables might be:
- supplied by server (spirit of RFC 7320; compression)
- supplied by client

`http://shop.example.com/stock/{sku}`

Note that the SKU is server supplied (somewhere else)

`http://multiply.example.com/{x,y}`

Client supplies values for x and y
Or, in searches:

`http://search.example.com/{item}`

In Web, usually mediated by form instead of URI template; put in query part or POST body

Pointers, Paths

Addressing components of a representation

— Building Fragment Identifiers for structured data

— Can also be used in the path/query part of URL

Usually defined in a data representation format ecosystem (e.g., [RFC 6901] JSON Pointer, XML XPath)

Pointers, Paths: Some Terminology

Expressiveness:
- Pointer: Works for one specific instance
- Path: Works for many instances of a class

Selector can return multiple places
CSS Selectors, XPath, …

No hard boundaries between these

Strong traction towards Turing equivalence (e.g., XPath)

Pointers and Paths in JSON ecosystem

RFC 6901: JSON pointer

— syntax for navigating in a JSON document by key or index

— syntax for representation in fragment identifier component

`#/odmAction/Reset_Min_and_Max_Measured_Values`

JSON Path

Highly simplied form of XPath

```
$.store.book[0].title
$['store']['book'][0]['title']
```

keying, subscript, slice; union

escape to expressions in underlying scripting language

https://goessner.net/articles/JsonPath/

Moving beyond text form

CoRAL: CRIs

Can be less tied up by text-based baggage
Could provide separation of URL and URI uses
Could combine with forms to obviate URL templates

[insert further wishlist here]

(But still need whiteboard form)