ENTERPRISE KNOWLEDGE

Global Knowledge & Information Management Services

Resolving Uniform Resource Identifiers (URIs)

Bess Schrader

Organizations that are getting started with the Resource Description Framework (RDF) often ask "Why do all of the entity identifiers look like URLs? Are they supposed to actually go somewhere?" This can be difficult to answer without taking the time to explore the foundations of the semantic web and one if its core components: the Uniform Resource Identifier (URI). While URIs *can* act like functioning URLs, resolving to an actual webpage, they don't always have to. In the following sections, we'll explore the ins and outs of URIs: why you may want to consider resolvable URIs, how to design your URIs so that they *can* resolve, and how to set up the technical process to ensure your URIs direct users to the resources you want them to see.

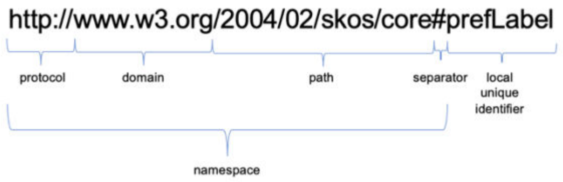
What is a URI?

URI stands for Uniform Resource Identifier. In RDF, a URI is the identifier used for an entity (also called a node, or object). Every entity in RDF **must** have a URI. A URI for an RDF entity is similar to the primary key for a row in a relational database table. However, unlike relational database identifiers, which are usually required to be unique only within a single table, URIs are intended to be **globally unique**. This means that they should not only be unique within your taxonomy or knowledge graph, but should be unique across *all* taxonomies and knowledge graphs, both at your organization and across the world.

But how can you possibly create a globally unique identifier without knowing all of the other URIs in existence? It's actually quite simple – use a URL! We do this with websites all of the time. You wouldn't want your website's unique identifier (its URL) to point to someone else's website, so you create a URL that uses a domain only you and your organization have access to. For example, Enterprise Knowledge owns the domain www.enterprise-knowledge.com. Because of this, we can be certain no one else in the world is building website URLs (or unique identifiers) that start with www.enterprise-knowledge.com, and therefore any URL we create on www.enterprise-knowledge.com is guaranteed to be **globally** unique.

The same concept applies with URIs. To ensure uniqueness, URIs typically look like URLs, starting with "http://". This gives you many more possibilities for creating unique identifiers than traditional database identifiers. For example, a row in a database table may have ID 1234, which is likely to be used again in a different table or database. A URI like https://enterprise-knowledge.com/author/bschrader is very unlikely to be reused in another knowledge graph.

So what does a URI look like? As previously established, URIs typically look like URLs, and the format of a URI is largely the same. Each URI has a protocol, a domain, a path, and an identifier that is unique within your domain and path:



Protocol

The protocol for a URI, like a URL, is almost always *http://* or *https://*

Domain

The domain is the "main page" of your URI, usually ending in .com, .org, .net, or .gov. Ideally, this piece of your URI corresponds to a server or webpage to which you have admin access or can register as a new domain with a web hosting service (more on this below). Just like with URLs, the www is not always required.

Path

The path of a URI can be as long or short as you want – in fact, you don't have to include a path at all. This can be human readable, providing breadcrumbs that drill down from the more general to more specific, or it can be opaque.

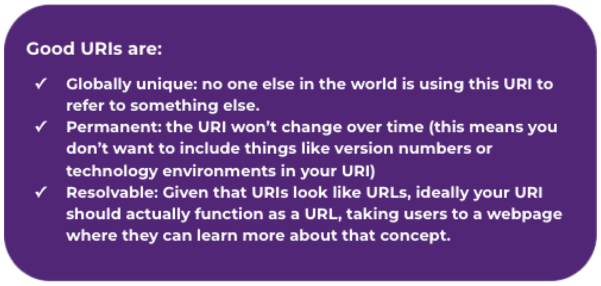
Separator

As the same suggests, the separator separates the protocol, domain, and path of your URI from the unique identifier. This separator is typically either the hash character (*#*) or a forward slash (*/*).

Together, the protocol, domain, path, and separator form a **namespace**. In the example above, the namespace is *http://www.w3.org/2004/02/skos/core#*.

Local Unique Identifier

Given that the URI itself is a unique identifier, it may seem strange to have a "unique identifier" component of the URI. However, this piece of the URI *only* needs to be unique **within the namespace**. For example, in the URI above, the local unique identifier is *prefLabel*. This means I can't reuse *prefLabel* in combination with the same namespace (*http://www.w3.org/2004/02/skos/core#*) to refer to a different RDF entity. However, I *can* reuse *prefLabel* with a *different* namespace to create a new, valid URI.


Good URIs are:

- ✓ **Globally unique: no one else in the world is using this URI to refer to something else.**
- ✓ **Permanent: the URI won't change over time (this means you don't want to include things like version numbers or technology environments in your URI)**
- ✓ **Resolvable: Given that URIs look like URLs, ideally your URI should actually function as a URL, taking users to a webpage where they can learn more about that concept.**

Adapted from *A data engineer's guide to semantic modelling* by Ilaria Maresi)

What Does it Mean for a URI to be Resolvable?

A resolvable, or dereferenceable, URI is a URI that actually points to a functioning website. For example, consider SKOS (the Simple Knowledge Organization System), an RDF framework typically used to create taxonomies. SKOS has several properties, such as preferred labels, alternate labels, and definitions. Each of these properties has its own URI:

http://www.w3.org/2004/02/skos/core#prefLabel
http://www.w3.org/2004/02/skos/core#altLabel
http://www.w3.org/2004/02/skos/core#definition

These URIs are all resolvable, which means clicking on any of these URIs takes you directly to the section of the SKOS documentation that defines that particular property, giving you more information about the topic.

Similarly, the popular open knowledge graph Wikidata provides resolvable URIs as well. Try clicking on the following URIs for various concepts to find all of the information Wikidata has for these entities:

Washington, D.C.: https://www.wikidata.org/entity/Q61
Enterprise Knowledge: https://www.wikidata.org/entity/Q91151007
The United Nations: https://www.wikidata.org/entity/Q1065
Tim Berners-Lee: https://www.wikidata.org/entity/Q80

Resolvable URIs allow you to easily share information about your taxonomies, ontologies, and knowledge graphs to users, both within your organization and (if desired) outside of your organization.

(continued on next page)

Why Should You Use Resolvable URIs?

Although they look like URLs, URIs are not *required* to resolve. For example, http://www.my-really-great-knowledge-graph.com/basketball is a valid URI, but if you plug it into a web browser, it won't actually go anywhere – it doesn't (at least at the time of writing this) resolve. However, it's highly recommended that the URIs you use actually *do* actually resolve (i.e. they point to a working webpage). The benefits of this are twofold:

Reducing Ambiguity and Providing Context

One of the core goals of the semantic web is to encode meaning in a machine readable way, thereby reducing ambiguity and adding meaning, or context, to data. A resolvable URI takes you to a page where you can learn more about that entity, meaning anyone that encounters your URI "in the wild" and wants to understand what that URI means can simply plug that URI into a web browser and visit the page it points to, giving them context and meaning around your URI.

Similarly, any machine, application, or process that encounters your URI in the wild can attempt to load the resource that your URI points to. If your URI resolves to a page that hosts actual RDF data, the machine that follows your URI can process this URI to actually understand what your URI is (what its type is, how it relates to other RDF entities, etc.).

By creating resolvable URIs, you're furthering the whole idea of *semantics* – reducing ambiguity and making meaning both human and machine readable.

Ensuring Global Uniqueness

A resolvable URI ensures that your URI is globally unique in the same way that URLs are globally unique. If I own a URL and put a webpage there, no one else can possibly use that URL to host a different page. The same is true for URIs. If I host a page of information about a concept at a URI address, no one else can use that URI to refer to a different concept. (While there's nothing *technically* stopping someone else from re-using this URI in a knowledge graph they create, they would be unable to have that URI resolve to anything other than my URI page, which would alert them that this URI is already in use. It's somewhat equivalent to someone else signing up for an account with your email address – they can do it, but they'll be unable to receive messages or reset their password, so it doesn't make much sense.)

Where Should Your Resolvable URIs Go?

If you want to create resolvable URIs, you'll need to determine what information should actually be shown to users when they click on a given URI. Do you want to show raw RDF data for an entity? Or maybe repackage that RDF data in a prettier HTML format, like the Wikidata and SKOS URIs above? Or do you want to direct to a page that doesn't incorporate the RDF at all, and instead shows totally human readable content? Should each entity have its own webpage, as in Wikidata? Or should you have a single web page where every entity gets its own *section* of that page? This is up to you to decide, based on your needs and the goals of your organization.

Generating the content for a resolvable URI page can be achieved through a variety of methods. Showing raw RDF data simply requires you to point to a text file – no web development required. For example, see the dereferenceable URI https://www.w3.org/2000/01/rdf-schema#label. While less human friendly, this is a viable option that provides detailed information about your URIs.

However, if you want to show something a bit more human-friendly, there are several options. Many taxonomy and ontology editors on the market today will automatically create concept web pages for taxonomy concepts, or HTML ontology documentation for ontologies, based on the information you add to your taxonomy or ontology. This can reduce the content-creation burden and streamline the process of creating resolvable pages.

Several open source tools, such as Widoco, can also be used to generate HTML ontology documentation for OWL compliant ontologies.

Of course, you always have the option of creating your own custom page for each URI you want to resolve. This can be time consuming, but provides you with greater flexibility and control of the information that is shown for each URI.

What Do You Need to Consider When Creating Your URIs?

Remember: URIs should be **permanent**. You don't want to change them. This means that choosing your URI structure carefully is a crucial task. Here are some key considerations when choosing how to create URIs:

What URL Domains Do You Have Access To?

If you want your URIs to be resolvable, you need to start them with a domain that you have access to. For example, if I choose to start all of my URIs with http://google.com, I will likely create valid URIs, but unless I work for Google, it's unlikely that I'll ever be able to host content on Google related to my URIs. Instead, I should start my URI with a domain that I own or have access to (for example: https://www.enterprise-knowledge.com/).

But what if you don't' have access to *any* domains?

If you don't have access to a web domain where you can either host your content or set up redirect links (more on this below), you can work with the W3C's Permanent Identifiers for the Web project. This project allows you to create URIs that begin with *https://w3id.org/*, and register your own redirects that they will host and serve, pointing your *https://w3id.org/* URIs wherever you choose. While this limits the flexibility you can have in your URI formats, this is a great option to ensure that you'll always be able to redirect and resolve your URIs without requiring you or your organization to maintain web servers.

(Continued at top)

Human Readable vs. Opaque URIs

Human readable URIs, as the name suggests, allow human beings to be able to make an educated guess as to what the URI refers to, simply by looking at the URI itself. For example, you may gather that the URI https://enterprise-knowledge.com/author/bschrader refers to an author who has the username *bschrader*. This is a very user-friendly approach, and has the added benefit that many webpages are also organized using human-readable URLs. For example, a human being reading the URL above could likely (and correctly) guess that there's a page about *bschrader* in the author section of enterprise-knowledge.com. If you want your URIs to resolve within an existing web page structure, human readable URIs may be a good option.

However, remembering that good URIs should be **permanent**, human-readable URIs may have some drawbacks. What if *bschrader* changes their name or username, and now goes by *bsmith*? You don't want to change the URI, but it may now become more confusing to users. Similarly, if *bschrader* transitions from being an author to, say, an editor, the URI can't change to reflect that.

Opaque URIs, on the other hand, do not allow a user to surmise what the URI might refer to. The wikidata URIs listed in the section above are a good example: the URI https://www.wikidata.org/entity/Q1065 gives no indication to a human being that this refers to the United Nations (until a human being visits the resolvable link, that is). While this can be less user-friendly, particularly for developers that may be querying an unfamiliar knowledge graph, opaque URIs are much more likely to be permanent. If the name of the United Nations changes at some point in the future, the Wikidata URI is not at all impacted. Opaque URIs are also a great option for multilingual taxonomies, ontologies, and knowledge graphs. By using an opaque identifier, you don't have to choose which language label is used in the URI, thereby giving equal footing to all languages used.

Both human-readable and opaque URIs are widely and successfully used across the semantic web. Whether you use human-readable or opaque URIs ultimately depends on your needs as an organization.

Slashes vs. Hashes

You may have noticed that the URI examples for SKOS and Wikidata in the section above use slightly different formats. Wikidata URIs use a slash to separate the wikidata prefix from the identifier for a concept (https://www.wikidata.org/wiki/Q1065), while SKOS URIs incorporate a hash to do the same (http://www.w3.org/2004/02/skos/core#prefLabel). Both of these are valid URI formats, and which one you decide to use depends on the type of page you want to use for URI dereferencing.

Slashes

Slashes are a good option when you want every URI to point to its **own** webpage. For example, visiting the UN URI (https://www.wikidata.org/wiki/Q1065) takes you to a page that *only* contains information about that UN concept. Slashes are a great option for URIs that talk about concepts, or individual entities in your knowledge graph.

Hashes

Hashes, on the other hand, take you to a specific **section** of a webpage. For example, the SKOS URIs below all take you to the same page, but to different sections, or anchor tags, within that page:

http://www.w3.org/2004/02/skos/core#prefLabel
http://www.w3.org/2004/02/skos/core#altLabel
http://www.w3.org/2004/02/skos/core#definition

If you plan to host a single webpage split into sections that define different URIs, hashes are a good option for your URI format. Hashes are often (but not always) used in URIs that define ontology elements (like classes, relationships, and attributes), enabling you to create a single webpage with your ontology documentation while still directing users to the specific section that defines an individual relationship or attribute URI.

Who Should See Your Resolvable Pages?

When creating pages to display content for resolvable URIs, you'll want to consider *who* should be able to see this content. Some organizations may want the general public to be able to see this information – this is particularly true for externally facing resources like public taxonomies or linked open data projects. In this case, you simply need to make sure that the website on which you plan to host the resolvable URI pages is publicly accessible. For example, https://enterprise-knowledge.com is publicly accessible, so anyone can access the content at https://enterprise-knowledge.com/author/bschrader.

However, in other cases, you may want your information to be accessible only to users inside your organization. In these instances, it's important to make sure that the pages to which your URIs point are either located behind your organization's firewall (i.e. only accessible to users on your organization's network or VPN), or that these pages require users to authenticate with some kind of organizational credentials. While this may impact your choice of URI, it is possible to have your URIs use a publicly accessible domain (like https://enterprise-knowledge.com), but redirect to a webpage that is controlled behind a firewall or via login (see below).

The Technical Nitty-Gritty: How Do You Make Your URIs Resolve?

So you've thought through your URI format and you've decided which content should be shown for each resolvable URI: now how do you get the URI to actually point to the content? First, in almost all cases, **you must have access to the domain for your URIs**. If you don't have access to a domain, you may want to consider working with the W3C's Permanent Identifiers for the Web project, as described above. Whether you're hosting content directly on a website in that domain, or redirecting URIs to point to a webpage hosted elsewhere, you'll need access to that domain to set up either option.

Assuming you have access to your URI domain, you have two main options:

How the Content at the Actual URL Location of the URI

This requires you to develop your URIs and your webpage together. For example, if you want to host content for the URI https://enterprise-knowledge.com/author/bschrader at that exact same URL, you need to:

1. have a webpage hosted at https://enterprise-knowledge.com
2. have an authors section on that webpage
3. host an HTML page for *bschrader* in the authors section of https://enterprise-knowledge.com

This approach also requires you to maintain the same structure of your web page in perpetuity. If the website is reorganized, and the authors section moves under a new parent page called "creators", the content for your URI may no longer be reachable

Create URL Redirects

This option is more flexible, and allows you to change the content to which your URIs point over time, without needing to change the URIs themselves. In this option, you'd need to access the server hosting the main webpage on your domain. For example, if my URIs all start with https://enterprise-knowledge.com, I'd need access to the server hosting that web page. Any server hosting a web page will be running a web server, like NGINX or Apache. In addition to serving web pages, these web servers have the ability to reroute traffic coming into the domain using URL redirects. For example, say you want any URI using the format

https://enterprise-knowledge.com/person/{username}

to actually point to a different page (including a website hosted elsewhere), such as:

https://my-really-great-knowledge-graph.com/person/{username}

You can set up a URL redirect in the proxy settings for your web server. As mentioned above, this redirect process can point URIs to other locations on the same domain, or to locations on a totally separate domain. For more details on how to set up these redirects, consult the documentation for your web server.

Conclusion

Resolvable URIs are a great way to make sure your semantic resources are both human and machine readable, improving access to knowledge and reducing ambiguity. Choosing good URIs that will be permanent, globally unique, and resolvable involves several considerations, but when done correctly, it can enable others to easily explore your semantic resources. To get help setting up resolvable URIs, contact us today