

Diverted URI Pattern

David Wood and James Leigh, 3 Round Stones Inc.
{jleigh, dwood}@3roundstones.com

Introduction

This paper develops a pattern for the optional local resolution of Linked Data URIs with remote HTTP authorities. This pattern allows URI identifiers from one authority to be mirrored, resolved and visualized by other authorities. Supporting use cases and a brief discussion of extent implementations are presented.

Intent

Resolve a data URI to the most appropriate authority in the context of the request. Prefer a resolution to a service capable of rendering human- and machine-readable views of the resource, even if the authority of the URI differs from the authority of the rendering service.

Motivation

Linked Data is often mirrored for the purposes of creating visualizations of the data, merging some or all of the data with data from other sources and/or enhancing responsiveness to queries. Because data resources are identified using URIs, resources that are mirrored to a machine using a different HTTP authority (consisting of a protocol scheme, a DNS machine name and an optional port number) [HTTP] are not generally directly resolvable on a server performing mirroring.

Web services that mirror Linked Data often have a requirement to serve information regarding resource URIs, regardless of whether they share an authority with those URIs. This is true, for example, when a service mirrors Linked Data for the purposes of providing a human-centric user interface for such data.

Linked Data “mash-ups” often share a problem with human-readable Web pages that are generated from underlying data: The data dies in the browser. By that we mean that the data used to render a human-centric view of the data is no longer accessible for reuse. The pattern described in this paper may be used to facilitate such data reuse by allowing the exposure of both human- and machine-readable data from authorities not represented in data URIs.

Requirements

Throughout this document example URI path components such as /go or /diverted may be replaced with any other string. They serve as identifiers for URI rewriting and routing only and thus represent implementation decisions.

Routing a URI to a Handler

A service attempting to serve information about resources identified by (possibly non-local) URIs must find a way to rewrite those URIs and route them as appropriate. This problem becomes somewhat exacerbated in a data-driven environment, because the service may not have *a priori* knowledge regarding the state of metadata associated with a particular URI. Thus, the service may need to check such state (e.g. via a database query) before determining appropriate routing.

The Diverted URI Pattern requires the routing of a URI to a handler to determine whether an authority switch is required.

Table 1 lists four cases of metadata state involving URIs and demonstrates the action to be taken in each case. Note that the second example (“URI is associated with a view template; the URI’s authority matches the serving host”) serves as an example of routing to a particular type of template for presentation to humans.

In each case, a URL redirection and rewriting process (called here the “go service”) evaluates both the URI presented to the service and the state of knowledge associated with that URI to determine the final outcome.

Table 1. URI Routing to Appropriate Authorities

Local Knowledge	Action Taken	Example URL Redirection
URI is unknown; it is not represented in the local RDF store	The go service redirects to the URI.	http://example.com/go?q=http://dbpedia.org/resource/Linked_Data -> http://dbpedia.org/resource/Linked_Data
URI is associated with a view template; the URI’s authority matches the serving host	The go service redirects to the URI with “?view” appended.	http://example.com/resource/Exelon -> http://example.com/resource/Exelon?view
URI is associated with a view template; the URI’s authority matches the serving host; the URI has a fragment id	The go service redirects to a diverted URI. NB: This case is most often used to allow for the presentation of human views of schemas using so-called “hash” URIs.	http://example.com/go?q=http://example.com/schema%23MyClass -> http://example.com/diverted;http%3A%2F%2Fexample.com%2Fschema%23MyClass
URI is associated with a view template; the URI’s authority does not match the serving host	The go service redirects to a diverted URI.	http://example.com/go?q=http://dbpedia.org/resource/Exelon -> http://example.com/diverted;http%3A%2F%2Fdbpedia.org%2Fresource%2FExelon

Handling a URI with a Different Authority

The “go service” described above presupposes a mechanism for resolving a request for a resource URI to a specific authority when the resource URI’s authority does not match the hosting service’s authority. A mechanism to handle that case is to embed a URL-encoded version of a remote URI into a URI that will resolve to the desired authority.

Table 2 illustrates the two possible cases; one in which a remote URI is embedded in a local URI and the other in which no such embedding is necessary.

Table 2. Diverted URIs

URI	Sample Served URL on example.com
http://dbpedia.org/resource/Exelon	http://example.com/diverted;http%3A%2F%2Fdbpedia.org%2Fresource%2FExelon
http://example.com/resource/Exelon	http://example.com/resource/Exelon

Existing Implementations

The Callimachus Project [Callimachus] implements the concepts presented in this paper. Similar approaches to satisfying the use cases are reportedly implemented in commercial products from Revelytix (<http://revelytix.com>) and Top Quadrant (<http://topquadrant.com>).

Consequences

The Diverted URI Pattern has the following benefits and liabilities:

1. URIs containing arbitrary HTTP authorities may be evaluated and contextually resolved by a Web service.
2. The combination of a “go service” and a diverted URL prefix allows both HTTP and non-http URIs to be resolved (e.g. DOI or URN URIs).
3. A “go service” following the Diverted URI Pattern is not limited to a specific authority. Such a service may redirect to external rendering services using the diverted prefix, thus further reducing the coupling between data URI and rendering services.
4. Simple URI resolution is diverted for the purpose of making routing and resolution decisions. This involves (possibly substantial) processing at a service host that is in addition to normal URI resolution.
5. URIs require rewriting to be resolvable to an authority of choice.

The Diverted URI Pattern bears some resemblance to the Objected Oriented pattern known as “Chain of Responsibility” [DesignPatterns]. Like Chain of Responsibility, Diverted URIs

reduces the coupling between an identifier and its ultimate handlers and adds flexibility in assigning responsibilities to handlers.

References

[Callimachus] The Callimachus Project. <http://callimachusproject.org>.

[DesignPatterns] Gamma, E., Helm, R., Johnson, R. and Vlissides, J. (1995). Design Patterns. Addison Wesley, New York, pp. 223–232.

[HTTP] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P. and Berners-Lee, T. (1999). Hypertext Transfer Protocol -- HTTP/1.1. Internet Engineering Task Force, Network Working Group, Request for Comments 2616. Retrieved 24 October 2011 from <http://www.w3.org/Protocols/rfc2616/rfc2616.html>.