Linked Data, Labels, URIs

Trey Terrell – Princeton University

QUICK RDF INTRODUCTION

By Pelf at en.wikipedia [Public domain], via Wikimedia Commons

Triples

<http://id.loc.gov/authorities/subjects/sh88003454>

<http://www.w3.org/2004/02/skos/core#prefLabel>

"Hydra (Greek mythology)"@en

Subject

The thing you're describing. Use a HTTP URI to represent it so you can use the web to get more information about it.

<http://id.loc.gov/authorities/subjects/sh88003454>

Predicate

What you're describing. Use a URI as an identifier so you can get more detailed information using the web.

<http://www.w3.org/2004/02/skos/core#prefLabel>

Object

The value of your description. Can be anything, including a URI.

Hydra (Greek mythology)"@en

Content Negotiation

curl -H "Accept: text/plain"
http://id.loc.gov/authorities/subjects/sh88003454 –L

Great, now what?

The Use Case

1. When entering data for a field in Hydra, I want to be able to put in RDF URIs instead, so that I can have a reliable identifier for a concept with extra metadata.

2. When displaying this value to the user I want to use the URI's preferred label, which is returned when I content negotiate.

Hydra Editor

- Gem for standardizing ingest forms. Used by Sufia, Curation Concerns, Oregon Digital, and others.
- Goal: Customize to allow for URIs to be entered.

Descriptions

* indicates required fields

Content

| Choose File | color.tif |

Title

| | + Add |

Alternative

| | + Add |

Convert Strings to URIs

- On ingest, convert all fields that are like "http://" to RDF::URIs.

```
1  class RecordsController < ApplicationController
2      include RecordsControllerBehavior
3
4      protected
5
6      def collect_form_attributes
7          AttributeURIConverter.new(super).convert_attributes
8      end
```

URI converter can be found at

https://github.com/OregonDigital/oregondigital_2/blob/master/app/services/attribute_uri_converter.rb

New Problem: Edit

- Now we've ingested a URI, but when we edit we see an ActiveTriples::Resource instead.

Title

#<ActiveTriples::Resource:0x007fbce82e83b0>   – Remove

+ Add

New Input for Hydra Editor

- Hydra Editor needs to be able to handle ActiveTriples::Resources

```ruby
class UriMultiValueInput < MultiValueInput
  protected

  def build_field(value, index)
    options = build_field_options(value, index)
    OregonDigital::URIEnabledStringField.new(@builder, attribute_name, options).field
  end
end
```

- https://github.com/OregonDigital/oregondigital_2/blob/master/lib/oregon_digital/uri_enabled_string_field.rb

New Problem: Get Label

Seeing the URI isn't helpful – what's it mean?

How to Get External Metadata

- Multiple options:
  - RDF.rb's RDF::Graph.load + RDF::Repository
    - Whenever you want new metadata, just load the URI and persist it somewhere. ActiveTriples supports this via ActiveTriples::Resource#fetch.

  - Marmotta + LDCache.
    - Marmotta (a triple-store) has the ability to automatically cache external triples if you request a URI from it.

  - CURL + LDP Cache (Fedora possible).
    - One could build a service that sends CURL requests and caches them in an LDP service.

External Metadata Goal

- Applied Linked Data group working on an abstract Linked Data Fragment solution for a service with pluggable backends for caching metadata.

- https://github.com/ActiveTriples/linked-data-fragments.git

Marmotta Option

- Tie Marmotta.rb (https://github.com/terrellt/marmotta) into ActiveTriples

```ruby
class TriplePoweredResource < ActiveTriples::Resource
  property :preflabel, :predicate => RDF::SKOS.prefLabel
  def repository
    @repository ||= MarmottaRepository.new(rdf_subject)
  end
end
```

Marmotta Option

```ruby
class MarmottaRepository
  attr_reader :uri
  def initialize(uri)
    @uri = uri
  end

  def query(*args)
    marmotta_resource.get
  end

  def delete(*args)
    marmotta_resource.delete
  end

  def <<(stuff)
    graph = marmotta_resource.get
    graph << stuff
    marmotta_resource.post(graph)
    true
  end

  private

  def marmotta_resource
    @marmotta_resource ||= Marmotta::Resource.new(uri, connection: OregonDigital.marmotta)
  end
end
```

Marmotta Option

```
1] pry(main)> t = TriplePoweredResource.new
> #<TriplePoweredResource:0x3ff911ceb16c(#<TriplePoweredResource:0x007ff2239d62d8>)>
2] pry(main)> t = TriplePoweredResource.new("http://id.loc.gov/authorities/subjects/sh88003454")
> #<TriplePoweredResource:0x3ff911c98fac(#<TriplePoweredResource:0x007ff223931f58>)>
3] pry(main)> t.rdf_label
> ["Hydra (Greek mythology)"]
4] pry(main)>
```

```
# Titles
property :title,
    :predicate => RDF::DC.title,
    :class_name => TriplePoweredResource
```

Title

Hydra (Greek mythology)	– Remove
	+ Add

Indexing

- URI in Fedora ✓

- Cached Metadata in Marmotta ✓

- Discoverable Metadata ✗

Indexing Challenges

1. Getting metadata may take a while.
2. Endpoints may be down.
3. Have to pull down the Fedora object to reindex.

Indexing Solutions

1. Index in the background.

2. Use Solr Atomic updates to avoid Fedora entirely.

3. Bonus – this allows for quick reindexing for new enrichments

Indexing in Action

- Current implementation sits in Oregon Digital:
  - Take in a solr document, iterate over URIs, run enrichments on them, and send atomic updates to solr.
  - https://github.com/OregonDigital/oregondigital_2/pull/107
- Applied Linked Data Group intends to create a standard side-car indexing routine for doing this both in and out of Hydra.

DEMO TIME!

What's Next?

We have the graph at our disposal. Some ideas:

1. Index skos#altLabels for discovery - search for "slipskin" and get "mandarins"
2. Index skos#narrower terms with a lower weight. Search for "sports", get "baseball"
3. Automatically build labels and tooltips for field identifiers. Hover over "Title" and get a tooltip pulled from the RDF for DC#title.
4. Use RDF sub-property relationships to automatically group facets by their lowest common denominator.
5. Use the language tags which are part of the RDF literals to automatically translate your facets.

Let's Use the Graph!

Questions?

Trey Terrell – Princeton University

Twitter: @terrell_dt

Email: tterrell@princeton.edu

Github: http://github.com/terrellt

Work done as a part of Oregon Digital, a collaboration between Oregon State University and the University of Oregon. Special thanks to everyone involved.