Currency Conversion
the Linked Data Way

SALAD 2013 Workshop, ESWC 2013
May 26, 2013, Montpellier, France

Alex Stolz and Martin Hepp
{firstname.lastname}@unibw.de
Motivation

Converting price values from one currency to another is a much-needed functionality in business applications that could be built on Linked Data

- Shopping comparison websites and product search engines
- B2B across different countries and between different business parties
- Portals for events, tickets, or services
- etc.
Motivation

Hotel comparison website at Web scale needs to compile a list of the $n$ (three) cheapest hotel room offers relative to the currency ($) preferred by the user.
Problem

- Prices are indicated in different currencies ($, €, ...)
- SPARQL processors provide no standard means to integrate currency conversion APIs into operations over RDF data
- Typical workflow:

<table>
<thead>
<tr>
<th>price</th>
<th>currency</th>
<th>price</th>
<th>currency</th>
</tr>
</thead>
<tbody>
<tr>
<td>89.0</td>
<td>EUR</td>
<td>42.9</td>
<td>EUR</td>
</tr>
<tr>
<td>90.0</td>
<td>USD</td>
<td>43.9</td>
<td>EUR</td>
</tr>
<tr>
<td>83.1</td>
<td>GBP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>42.9</td>
<td>EUR</td>
<td>69.1</td>
<td>EUR</td>
</tr>
<tr>
<td>43.9</td>
<td>EUR</td>
<td>89.0</td>
<td>EUR</td>
</tr>
<tr>
<td></td>
<td></td>
<td>97.6</td>
<td>EUR</td>
</tr>
</tbody>
</table>

Problematic, if SPARQL query returns partial result set (5 results out of 100)
Exchange Rates for Linked Data

RESTful Web service for the delivery of currency exchange rates in RDF, that

- adheres to the Linked Open Data (LOD) design principles
- removes the need for proprietary code
- can be accessed from client-side JavaScript
- works with any standard SPARQL processor able to retrieve RDF by dereferencing a resource URI

http://www.currency2currency.org/
Conceptual Overview

Currency Conversion API

Exchange Rates for LOD

Request update

Data Store

Cache

XRO

RDF

Currency Conversion API

http://www.currency2currency.org/
Exchange Rate Ontology (XRO)

http://purl.org/xro/
Currency Exchange Rate (EUR→USD) in N3

@prefix dbpedia: <http://dbpedia.org/resource/> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xch_EUR: <http://www.currency2currency.org/EUR#> .
@prefix xro: <http://purl.org/xro/ns#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

xch_EUR:USD a xro:ExchangeRateInfo;
  rdfs:label "Euros to U.S. dollar"@en;
  rdfs:comment "1 EUR = ? USD"@en;
  xro:base dbpedia:Euro;
  xro:counter dbpedia:United_States_dollar;
  xro:rate "1.31010"^^xsd:decimal;
  xro:inverseRate "0.7633010"^^xsd:decimal;
  dcterms:source <http://www.google.com/ig/calculator?hl=en&q=1EUR=?USD>;
  xro:timeOfConversion "2013-04-11T00:00:02Z"^^xsd:dateTime .

base and counter currency instances from DBPedia are obtained using dbprop:isoCode
Calculating Exchange Rates

Google Calculator service:
- http://www.google.com/ig/calculator?hl=en&q=1USD=?EUR
- http://www.google.com/ig/calculator?hl=en&q=1USD=?GBP
- ...

For \( n \) currencies, only \( n \) requests are necessary in order to compute all possible exchange rate combinations

- USD\( \rightarrow \)EUR is inverse of EUR\( \rightarrow \)USD
- EUR\( \rightarrow \)GBP is EUR\( \rightarrow \)USD\( \rightarrow \)GBP (transitive)

\[ rate_{A2B} = \frac{rate_{A2BASE}}{rate_{B2BASE}} \]
URI Patterns

- Cool URIs
  - http://www.currency2currency.org/<base>/<counter>
  - http://www.currency2currency.org/<base>
  - http://www.currency2currency.org/

- Timestamps (archiving functionality):
  - <uri>/YYYYMMDD

- Entity URIs for exchange rates:
  - E.g. http://www.currency2currency.org/USD#EUR

*base and counter currencies represented by 3-letter currency codes according to the ISO 4217 standard, e.g. EUR, USD, ...*
Content Negotiation

$ curl –H “Accept: text/n3;q=1.0” http://www.currency2currency.org/CHF

HTTP/1.1 200 OK
Content-Location: http://www.currency2currency.org/CHF
Access-Control-Allow-Origin: *
Vary: Accept
Cache-Control: max-age=3600, must-revalidate
Content-Type: text/n3
Content-Length: 31602
Date: Thu, 11 Apr 2013 10:31:55 GMT

Allow for client-side (JavaScript) cross-origin requests using CORS-header:

Access-Control-Allow-Origin: *
## Content Negotiation

<table>
<thead>
<tr>
<th>Serialization format</th>
<th>Media types accepted (Accept:)</th>
<th>Content type delivered (Content-Type:)</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTML</td>
<td>not available text/html application/xhtml+xml</td>
<td>text/html</td>
</tr>
<tr>
<td>RDF/XML</td>
<td>application/rdf+xml application/xml</td>
<td>application/rdf+xml</td>
</tr>
<tr>
<td>N3</td>
<td>text/n3 text/rdf+n3 application/n3</td>
<td>text/n3</td>
</tr>
<tr>
<td>Turtle</td>
<td>text/turtle application/x-turtle</td>
<td>text/turtle</td>
</tr>
<tr>
<td>RDF/JSON</td>
<td>application/json text/rdf+json text/javascript</td>
<td>application/json</td>
</tr>
<tr>
<td>N-Triples</td>
<td>text/plain</td>
<td>text/plain</td>
</tr>
</tbody>
</table>
Caching

- http://www.currency2currency.org → File store
- http://www.currency2currency.org/EUR → Memcache
- http://www.currency2currency.org/EUR/USD → Client cache

<table>
<thead>
<tr>
<th></th>
<th>Data store</th>
<th>Client cache</th>
<th>Memcache</th>
<th>File store²</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Type</strong></td>
<td>NoSQL (schemaless) object data store</td>
<td>Local cache of client application</td>
<td>Distributed memory object caching system</td>
<td>Data object store for large files</td>
</tr>
<tr>
<td><strong>Expiry</strong></td>
<td>Never</td>
<td>After 1 hour</td>
<td>After 6 hours</td>
<td>Never</td>
</tr>
<tr>
<td><strong>Update frequency</strong></td>
<td>Daily</td>
<td>Cache lifetime expired</td>
<td>Cache limits or lifetime reached</td>
<td>Daily</td>
</tr>
<tr>
<td><strong>Memory limits</strong></td>
<td>Limited to storage capacity of used data type</td>
<td>Unlimited, application-dependent</td>
<td>1 Megabyte</td>
<td>Limited to data transfer limit</td>
</tr>
<tr>
<td><strong>Intended usage</strong></td>
<td>Daily updated list of exchange rates</td>
<td>Downloaded page contents</td>
<td>Small-sized, partial RDF serializations</td>
<td>Complete RDF dumps</td>
</tr>
</tbody>
</table>

Alex Stolz, alex.stolz@unibw.de
Adherence to Linked Data Guidelines

1. Every exchange rate entity obtains a named URI
2. Exchange rates can be looked up easily because relying on HTTP URIs
3. When someone looks up a URI of an exchange rate, useful information is displayed to humans and machines alike (content negotiation)
4. Links to other datasets, e.g. to currency instances of DBPedia, allow for discovering additional things
Example of a Currency Conversion in SPARQL

Simplifying assumptions:

- DBPedia currency instances are preloaded on the SPARQL endpoint
- Web shop data and exchange rates related to *U.S. dollar* are available

```
PREFIX gr: <http://purl.org/goodrelations/v1#>
PREFIX xro: <http://purl.org/xro/ns#>
PREFIX dbpedia: <http://dbpedia.org/resource/>
PREFIX dbpprop: <http://dbpedia.org/property/>

WHERE {
  ?s a gr:Offering; gr:hasPriceSpecification ?pspec .
  ?pspec gr:hasCurrency ?code; gr:hasCurrencyValue ?price .
  ?xrate xro:rate ?rate;
  FILTER(str(?counter_code) = str(?code))
}
ORDER BY ?base_price LIMIT 5
```
Tabular Results of SPARQL Query

<table>
<thead>
<tr>
<th>price</th>
<th>code</th>
<th>base_price</th>
<th>base_code</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>USD</td>
<td>0.0</td>
<td>USD</td>
</tr>
<tr>
<td>1.29</td>
<td>RON</td>
<td>0.38123885</td>
<td>USD</td>
</tr>
<tr>
<td>1.93</td>
<td>RON</td>
<td>0.5703806</td>
<td>USD</td>
</tr>
<tr>
<td>2.58</td>
<td>RON</td>
<td>0.7624777</td>
<td>USD</td>
</tr>
<tr>
<td>3.13</td>
<td>RON</td>
<td>0.92502147</td>
<td>USD</td>
</tr>
</tbody>
</table>

\[
price_{BASE} = \frac{price_A}{rate_{A2BASE}} = \frac{1.29}{3.382^*} = 0.381
\]

* Currency conversion based on USD exchange rates as of 10.04.2013: http://www.currency2currency.org/USD/20130410
Comparison with Related Work

Existing approaches for currency conversion on the Semantic Web:

- No RDF-based Web services
- Approaches with data models for expressing currencies
  - QUDT (for SPIN functions)
  - Dataset for exchange rates in the European LOD2 project (exposed as a SPARQL endpoint)
- Ontology for Units of Measure and Related Concepts (no currencies)

Our approach:

- Fully-fledged framework for currency conversions
  - Schema for exchange rates
  - Web service for RDF
  - JavaScript-friendly
  - Linked-Data-compliant
- Currency exchange rates in RDF, so SPARQL queries can take advantage
Future Extensions

- Conceptual improvements:
  - Extend Exchange Rate Ontology model by additional, domain-related properties and concepts (place of trade, type of transaction, type of market, etc.)

- Technical improvements:
  - Set up a RDF store with SPARQL 1.1 Query Federation capability to serve most up-to-date currency exchange rates
  - Make our service operable with different Web services (with respective provenance information to enhance trust aspect of service)
Conclusion

We proposed a RESTful Web service for currency exchange rates in RDF

- based on open Web APIs for currency conversion
- ensuring interoperability across diverse data sources on the Web of Data
- usable with SPARQL queries and standard SPARQL processors

Our service can serve as a generic pattern for integrating other open and dynamic Web APIs and making them available in the LOD cloud, such as unit conversion, product review data, weather information, etc.
Thank You!

Alex Stolz
alex.stolz@unibw.de

Project page:
http://www.currency2currency.org/
Acknowledgments

- This work has been supported by the German Federal Ministry of Education and Research (BMBF) by a grant under the KMU Innovativ program as part of the Intelligent Match project (FKZ 01IS10022B)

- Image Credits:
  - Creative Commons pictures from Flickr.com
  - http://www.w3.org/DesignIssues/LinkedData.html