W3C RDF Working Group

JSON Syntax Options

F2F Meeting 1, Amsterdam
13-14 April 2011

Thomas Steiner, Research Scientist, Google
tomac@google.com, @tomayac

From the group's charter

"Define and standardize a JSON Syntax for RDF [...]. The goal is to provide an RDF serialization as complete as possible [...], but features may be ignored and special syntax features may be introduced if that would greatly facilitate the adoption of the JSON encoding by the Web Application developers' community."

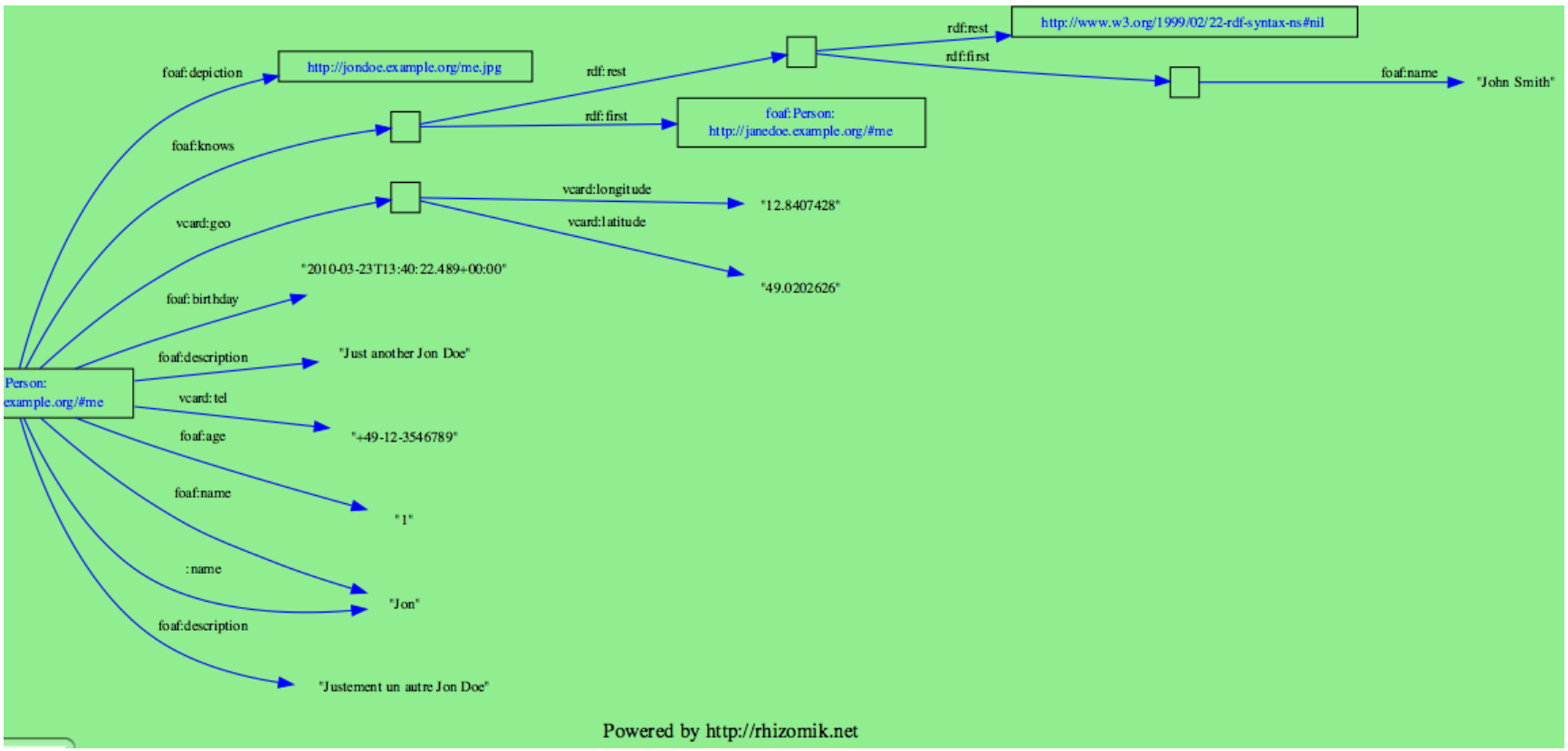
http://www.w3.org/2011/01/rdf-wg-charter#scope

Standing on the shoulders of giants

- RDF JSON by Talis
- JSON-LD by Manu Sporny
- JRON by Sandro Hawke
- JSON formatting in the Linked Data API
- Serializing SPARQL Query Results In JSON by DAWG
- JSN3 by Nathan Rixham
- Flat triples approach to RDF graphs by Dominik Tomaszuk
- Rdfj by Mark Birbeck
- JTriples by Michael Hausenblas

Sample graph

One shared sample graph with some interesting features
http://www.w3.org/2011/rdf-wg/wiki/JSON-Serialization-Examples#Shared_Example_for_Serialization_Lineup_.28Turtle.29



Sample graph

Artificial data set with some real-world issues:

- Literals
- Literals with language tags
- Literals with trivial data types
- Literals with non-trivial data types
- Blank nodes
- Lists
- IRIs
- Potentially colliding CURIE prefixes

Serialization lineup

See concrete serializations here:

http://www.w3.org/2011/rdf-wg/wiki/JSON-Serialization-Examples#JSON_Serializations_Lineup

In the following overall observations of the JSON proposals.

Object-based or triple-based

Object-based

The idea is to enable a holistic view (invented syntax):

```
{
  "name": "Jon Doe",
  "age": 29,
  "homepage": "http://jon.ex.org/"
}
```

This shall allow for direct access of properties (after JSON.parse() step):

```
obj.name
obj.homepage
```

Triple-based

The idea is to stay as close to the metal as possible (invented syntax):

```
[
  {"s": "#", "p": "name", "o": "Jon Doe"},
  {"s": "#", "p": "age", "o": 29},
  {"s": "#", "p": "homepage", "o":
     "http://jon.ex.org"}
]
```

Findings - subjects (object-based)

Introduce subjects as a new object level:

```
{
  "@subject": "_:id",
  "name": "Jon Doe"
}
```

vs.

```
{
  "_:id": {
    "name": "Jon Doe"
  }
}
```

Findings - IRIs

Differentiate between string literals and IRIs:

"http://ex.org" vs. "<http://ex.org>"

Mostly considered unnecessary.

Findings - language tags

Differentiate between string literals in different languages:

"hi" vs. "hi@en" vs. "salut@fr"

Mostly considered necessary.

- Microsyntax: "hi@en"
- Value pairs: {"value": "hi", "language": "en"}
- Key map: {"en": "hi"}

Findings - non-native types

Differentiate between non-native types (native JSON types are string, number, true, false, null, object, array):

2@int vs. 2@int32 vs. 2@int64

Mostly considered necessary.

- Microsyntax: "2@xsd:integer"
- Value pairs: {"value": 2, "type": "xsd:integer"}
- Key map: {"xsd:integer": 2}

Findings - lists

Reference lists:

```
{
  "knows": [{"name": "Jane"}, {"name": "Jim"}]
}
```

Mostly considered borderline.

- Array: [{"name": "Jane"}, {"name": "Jim"}]

Findings - namespaces

Differentiate between foaf:name and ex:name:

```
{
  "@prefixes": {"ex": "http://ex.com/ns/1/"},
  "ex:name": "Jane"
}
```

Mostly considered borderline / necessary.

- Prefix map: {"ex": "http://ex.com/ns/1/"}

Token map: {"name": "http://ex.com/ns/1/name"}

- Full IRIs: {"http://ex.com/ns/1/name": "Jane"}
- Well-known external default profile

Key take-aways

Do we need two pills? Red? Blue? Can there be one purple pill to rule them all?

Do we need language tags?

Do we need non-native types?

Do we need the more obscure RDF features at all (in JSON)?

Do we need mappings from vanilla JSON to RDFied JSON?

Credits, contact, thanks

Most (all?) of this covered before (and better) by Nathan Rixham in http://www.w3.org/2011/rdf-wg/wiki/JSON_Syntax_Options#JSON_Syntax_Options

Red pill blue pill image "fair use" of a still frame from the movie The Matrix

Some help from Hannes Mühleisen (@hfmuehleisen)

Thomas Steiner tomac@google.com, @tomayac