



Pasquale Lisenza
@pasqlisena

Easy Web API Development with SPARQL Transformer

...a simple use case

EXPLORE THE LOUVRE



St. John the Baptist [EN]
Saint Jean-Baptiste [FR]
San Giovanni Battista [IT]



Fishing [EN]



Combat of Love and Chastity [EN]
Le Combat de l'Amour et de la
Chasteté [FR]
Lotta tra Amore e Castità [IT]



Bonaparte Visiting the Plague
Victims of Jaffa [EN]
Bonaparte visitant les
pestiférés de Jaffa [FR]

...a simple use case

```
SELECT * WHERE {  
    ?work a dbo:Work ;  
        dbo:museum dbr:Louvre ;  
        foaf:depiction ?image ;  
        rdfs:label ?name .  
} LIMIT 100
```

SPARQL result JSON format

-- [W3C specification](#)

```
{  
  "head": {},  
  "results": {  
    "distinct": false,  
    "ordered": true,  
    "bindings": [{  
      "work": {  
        "type": "uri",  
        "value": "http://dbpedia.org/resource/San_Giovanni_in_Catona"  
      },  
      "image": {  
        "type": "uri",  
        "value": "http://en.wikipedia.org/wiki/San_Giovanni_in_Catona#/media/File:San_Giovanni_in_Catona.jpg"  
      },  
      "name": {  
        "type": "literal",  
        "xml:lang": "it",  
        "value": "San Giovanni in Catona"  
      }  
    }, {  
      "work": {  
        "type": "uri",  
        "value": "http://dbpedia.org/resource/Saint_Jean-Baptiste_(Leonardo_de_Vinci)"  
      },  
      "image": {  
        "type": "uri",  
        "value": "http://en.wikipedia.org/wiki/Saint_Jean-Baptiste_(Leonardo_de_Vinci)_%28DBpedia%29#/media/File:Saint_Jean-Baptiste_(Leonardo_de_Vinci).jpg"  
      },  
      "name": {  
        "type": "literal",  
        "xml:lang": "fr",  
        "value": "Saint Jean-Baptiste (Leonardo de Vinci)"  
      }  
    }  
  }  
}
```

```
{  
  "head": {},  
  "results": {  
    "distinct": false,  
    "ordered": true,  
    "bindings": [{  
      ...  
    }]  
  }  
}
```

metadata

solutions

from [DBpedia](#)



SPARQL result JSON format

-- [W3C specification](#)

RESULTS.BINDINGS (solutions)

```
"bindings": [{" "work": { "type": "uri", "value": "http://dbpedia.org/resource/St._John_the_Baptist_(Leonardo)" }, "image": { "type": "uri", "value": "http://...Saint_John_the_Baptist_C2RMF_retouched.jpg" }, "name": { "type": "literal", "xml:lang": "it", "value": "San Giovanni Battista (Leonardo)" } }, {" "work": { "type": "uri", "value": "http://dbpedia.org/resource/St._John_the_Baptist_(Leonardo)" }, "image": { "type": "uri", "value": "http://...Saint_John_the_Baptist_C2RMF_retouched.jpg" }, "name": { "type": "literal", "xml:lang": "en", "value": "St. John the Baptist (Leonardo)" } }, {" "work": { "type": "uri", "value": "http://dbpedia.org/resource/Fishing_(Carracci)" }, "image": { "type": "uri", "value": "http://.../Annibale_Carracci_-_Fishing_-_WGA4414.jpg" }, "name": { "type": "literal", "xml:lang": "en", "value": "Fishing (Carracci)" } }]
```

SAME DIFFERENT

SAME DIFFERENT

SAME DIFFERENT



from [Dbpedia](#)



```
[{
```

```
  "id": "http://dbpedia.org/resource/St._John_the_Baptist_(Leonardo)",  
  "name": [{  
    "language": "en",  
    "value": "St. John the Baptist (Leonardo)"  
  }, {  
    "language": "it",  
    "value": "San Giovanni Battista (Leonardo)"  
  }],  
  "image": "http://...Saint_John_the_Baptist_C2RMF_retoucheda.jpg"  
},  
{  
  "id": "http://dbpedia.org/resource/Fishing_(Carracci)",  
  "name": {  
    "language": "en",  
    "value": "Fishing (Carracci)"  
  },  
  "image": "http://.../Annibale_Carracci_-_Fishing_-_WGA4414.jpg"
```

2
names



1
picture



The princess so far: the developer!

skip irrelevant metadata

reducing and parsing

merging the “rows”

mapping to different structures

<https://github.com/w3c/EasierRDF>

Booth, D., Chute, C.G., Glaser, H., Solbrig, H.: **Toward Easier RDF**.
In: *W3C Workshop on Web Standardization for Graph Data*. Berlin, Germany (2019)



Different proposed solutions

Wikidata SDK

SPARQL Template
Transformation Language
(STTL)

HyperGraphQL

sparql-to-jsonld

JSON Schema

RDFJS

SOLID &
LDflex

CONSTRUCT

GrapQL-LD

None of them implement all the four tasks

skip irrelevant metadata

mapping to different structures

merging the “rows”

reducing and parsing



The new princess: **SPARQL Transformer**



[/D2KLab/sparql-transformer](https://github.com/D2KLab/sparql-transformer)

- A JSON-based query syntax
- A library in JavaScript and Python
- A new way of exposing web APIs

Query Object

KEY
CONCEPT

Define **at the same time**:

- the query
- the desired template

```
{  
  "proto": {  
    "id" : "?work",  
    "name": "$rdfs:label$required",  
    "image": "$foaf:depiction$required"  
  },  
  "$where": [  
    "?work a dbo:Work",  
    "?work dbo:museum dbr:Louvre"  
  ],  
  "$limit": 100  
}
```

PROTOTYPE DEFINITION

- write directly the resulting shape
- define the replacements
- options and filters

\$-PROPERTIES

map SPARQL features

\$where \$values \$limit \$distinct \$orderby
\$groupby \$having \$filter \$prefixes

Query Object

Objects are merged on an **anchor** (id)

```
"proto": {  
    "id" : "?work",  
    "name": "$rdfs:label$required",  
    "image": "$foaf:depiction$required",  
    "author": {  
        "id": "?author",  
        "name": "$rdfs:label$required"  
    }  
}, "$where": "?work dbo:author ?author"
```

?work rdfs:label ?name .
?work foaf:depiction ?image .

at different levels

?author rdfs:label ?name .

INPUT: JSON query

```
{  
  "proto": {  
    "id": "?id",  
    "name": "$rdfs:label$required",  
    "image": "$foaf:depiction$required"  
  },  
  "$where": [  
    "?id a dbo:City",  
    "?id dbo:country dbr:Italy"  
  ],  
  "$limit": 100  
}
```

OUTPUT: SPARQL query

```
SELECT DISTINCT ?id ?v1 ?v2  
WHERE {  
  ?id a dbo:City.  
  ?id dbo:country dbr:Italy.  
  ?id rdfs:label ?v1.  
  ?id foaf:depiction ?v2  
}  
  
LIMIT 100
```

Endpoint:

<https://dbpedia.org/sparql>[EXECUTE](#)

TRANSFORMED

ORIGINAL

[d2klab.github.io/
sparql-transformer](https://d2klab.github.io/sparql-transformer)

```
[  
  {  
    "id": "http://dbpedia.org/resource/Bologna",  
    "name": [  
      {  
        "language": "en",  
        "value": "Bologna"  
      },  
      {  
        "language": "it",  
        "value": "Bologna"  
      }  
    ]  
  }]
```

JSON-LD Version

```
{  
  "@context" : "http://schema.org/",  
  "@graph": {  
    "@id" : "?work",  
    "@type" : "Painting",  
    "name": "$rdfs:label$required",  
    "image": "$foaf:depiction$required"  
  },  
  "$where": [  
    "?work a dbo:Work",  
    "?work dbo:museum dbr:Louvre"  
  ],  
  "$limit": 100  
}
```

Query Object



```
{  
  "@context": "http://schema.org/",  
  "@graph": [  
    {"@id": "http://.../St._John_the_Baptist_(Leonardo)",  
     "@type": "Painting",  
     "name": [{"  
       "@language": "en",  
       "@value": "St. John the Baptist (Leonardo)"  
     }, {"  
       "@language": "it",  
       "@value": "San Giovanni Battista (Leonardo)"  
     }],  
     "image": "http://..._C2RMF_retouched.jpg"  
   },  
   ...  
 ]}  
}
```

JSON Output



grlc + SPARQL Transformer



1. Store your JSON queries on a **GitHub** repository
2. Automatically build a **Swagger documentation**
3. Use your **API** and **transform** your data

Meroño-Peña, A., Hoekstra, R.: **grlc Makes GitHub Taste Like Linked Data APIs.**
In: The Semantic Web – ESWC 2016 Satellite Events. Heraklion, Greece (2016)

Used in



SILKNOW



memad-api

[Base URL: grlc.eurecom.fr/api/Siliam/memad-api/]
[/api/Siliam/memad-api/spec](http://grlc.eurecom.fr/api/Siliam/memad-api/spec)

[Siliam - Website](#)

[License](#)

TV

GET /program_detail

GET /program_list List of programs.

Models

Message >

Oh, yeah?

Quantitative evaluation

QUERIES*	n. objects		time (ms)	
	original	transformed	original	transformed
1.Born_in_Berlin	1132	573	101	168
2.German_musicians	290	257	49	61
3.Musicians_born_in_Berlin	172	109	51	59
4.Soccer_players	78	70	203	210
5.Games	1020	981	70	121

Overhead < 0.1 seconds

* from <https://wiki.dbpedia.org/onlineaccess>

Quantitative evaluation



About: Adalberto di Prussia (1811-1873)

An Entity of Type : [persona](#), from Named Graph : <http://dbpedia.org>, within Data Space : [dbpedia.org](#)

Prince Adalbert of Prussia (Heinrich Wilhelm Adalbert) (29 October 1811 in Berlin – 6 June 1873 in Karlsbad) was a son of Prince Wilhelm of Prussia and Landgravine Marie Anna of Hesse-Homburg. He was a naval theorist and admiral. He was instrumental during the Revolutions of 1848 in founding the first unified German fleet, the Reichsflotte. During the 1850s he helped to establish the Prussian Navy. Adalbert was the son of Prince William, the youngest brother of King Frederick William III.

Property	Value
dbo:birthDate	<ul style="list-style-type: none">1811-10-29 (xsd:date)
dbo:deathDate	<ul style="list-style-type: none">1873-06-06 (xsd:date)1873-6-6
rdfs:label	<ul style="list-style-type: none">Prince Adalbert of Prussia (1811–1873) (en)Henri Guillaume Adalbert de Prusse (fr)Adalberto di Prussia (1811-1873) (it)Adalbert van Pruisen (1811-1873) (nl)
owl:sameAs	<ul style="list-style-type: none">dbr:Adalberto di Prussia (1811-1873)http://www.Adalberto di Prussia (1811-1873)

[http://dbpedia.org/resource/Prince
Adalbert of Prussia \(1811-1873\)](http://dbpedia.org/resource/Prince_Adalbert_of_Prussia_(1811-1873))



Qualitative evaluation

“How much do you prefer B respect to A?”

A. SPARQL standard JSON

B. Transformed JSON

User Survey

55 subjects from the Semantic Web Mailing list

A. User Interface (flat)

band	album	albumDate	member
http://dbpedia.org/resource/An_Albatross	http://dbpedia.org/resource/Blessphemy...of_the_Peace_Beast_Feastgiver_and_the_Bear_Warp_Kumite		http://dbpedia.org/resource/Porter_(band)
http://dbpedia.org/resource/An_Albatross	http://dbpedia.org/resource/Eat_Lightning_Shit_Thunder		http://dbpedia.org/resource/Porter_(band)
http://dbpedia.org/resource/Porter_(band)	http://dbpedia.org/resource/Atemahawke	"2007-05-15"^^xsd:date	http://dbpedia.org/resource/Porter_(band)
http://dbpedia.org/resource/Porter_(band)	http://dbpedia.org/resource/Atemahawke	"2007-05-15"^^xsd:date	http://dbpedia.org/resource/Porter_(band)
http://dbpedia.org/resource/Porter_(band)	http://dbpedia.org/resource/Atemahawke	"2007-05-15"^^xsd:date	http://dbpedia.org/resource/Porter_(band)
http://dbpedia.org/resource/Porter_(band)	http://dbpedia.org/resource/Porter_(band)	"2007-05-15"^^xsd:date	http://dbpedia.org/resource/Porter_(band)
http://dbpedia.org/resource/Porter_(band)	http://dbpedia.org/resource/Donde_las_ponys_pastan		http://dbpedia.org/resource/Porter_(band)
http://dbpedia.org/resource/Porter_(band)	http://dbpedia.org/resource/Donde_las_ponys_pastan		http://dbpedia.org/resource/Porter_(band)
http://dbpedia.org/resource/Porter_(band)	http://dbpedia.org/resource/Donde_las_ponys_pastan		http://dbpedia.org/resource/Porter_(band)
http://dbpedia.org/resource/Porter_(band)	http://dbpedia.org/resource/Donde_las_ponys_pastan		http://dbpedia.org/resource/Porter_(band)

B. User Interface (nested)

id	album	member												
http://dbpedia.org/resource/Caramelos_de_Cianuro	<table border="1"><thead><tr><th>id</th><th>date</th></tr></thead><tbody><tr><td>http://dbpedia.org/resource/Caramelos_de_Cianuro_(album)</td><td></td></tr><tr><td>http://dbpedia.org/resource/Flor_De_Fuego</td><td></td></tr><tr><td>http://dbpedia.org/resource/Frisbee_(album)</td><td></td></tr><tr><td>http://dbpedia.org/resource/Harakiri_City</td><td></td></tr><tr><td>http://dbpedia.org/resource/Miss_Mujerzuela</td><td>2000-08-22</td></tr></tbody></table>	id	date	http://dbpedia.org/resource/Caramelos_de_Cianuro_(album)		http://dbpedia.org/resource/Flor_De_Fuego		http://dbpedia.org/resource/Frisbee_(album)		http://dbpedia.org/resource/Harakiri_City		http://dbpedia.org/resource/Miss_Mujerzuela	2000-08-22	http://dbpedia.org/resource/Porter_(band)
id	date													
http://dbpedia.org/resource/Caramelos_de_Cianuro_(album)														
http://dbpedia.org/resource/Flor_De_Fuego														
http://dbpedia.org/resource/Frisbee_(album)														
http://dbpedia.org/resource/Harakiri_City														
http://dbpedia.org/resource/Miss_Mujerzuela	2000-08-22													
http://dbpedia.org/resource/Diva_Destruction	<table border="1"><thead><tr><th>id</th><th>date</th></tr></thead><tbody><tr><td>http://dbpedia.org/resource/Exposing_the_Sickness</td><td>2003-02-24</td></tr></tbody></table>	id	date	http://dbpedia.org/resource/Exposing_the_Sickness	2003-02-24	http://dbpedia.org/resource/Porter_(band)								
id	date													
http://dbpedia.org/resource/Exposing_the_Sickness	2003-02-24													

Qualitative evaluation

User Survey

55 subjects from the Semantic Web Mailing list

TYPE	LEVEL	I prefer the other solution			I prefer SPARQL Transformer	
		-2	-1	0	+1	+2
JSON results	0 (no nesting)	6	6	4	14	26
	1 (one nesting)	5	5	3	21	21
	2 (two nestings)	3	9	5	17	21
User Interface	0 (no nesting)	4	8	3	19	21
	1 (one nesting)	3	10	2	20	20
	2 (two nestings)	4	7	3	16	25

In summary

- Go straight to the data
- Un-pack values and parse
- Merge objects by “id”
- Reshape (also in JSON-LD)
- Web API with grlc
- Playground for fast testing

Applications

- API implementation
- Middleware for web app
- Fast prototyping for visualising the data

Future work

- SPARQL features like nested queries, UNION
- ASK, INSERT, DELETE
- Separate in 2 modules (querying and reshaping)

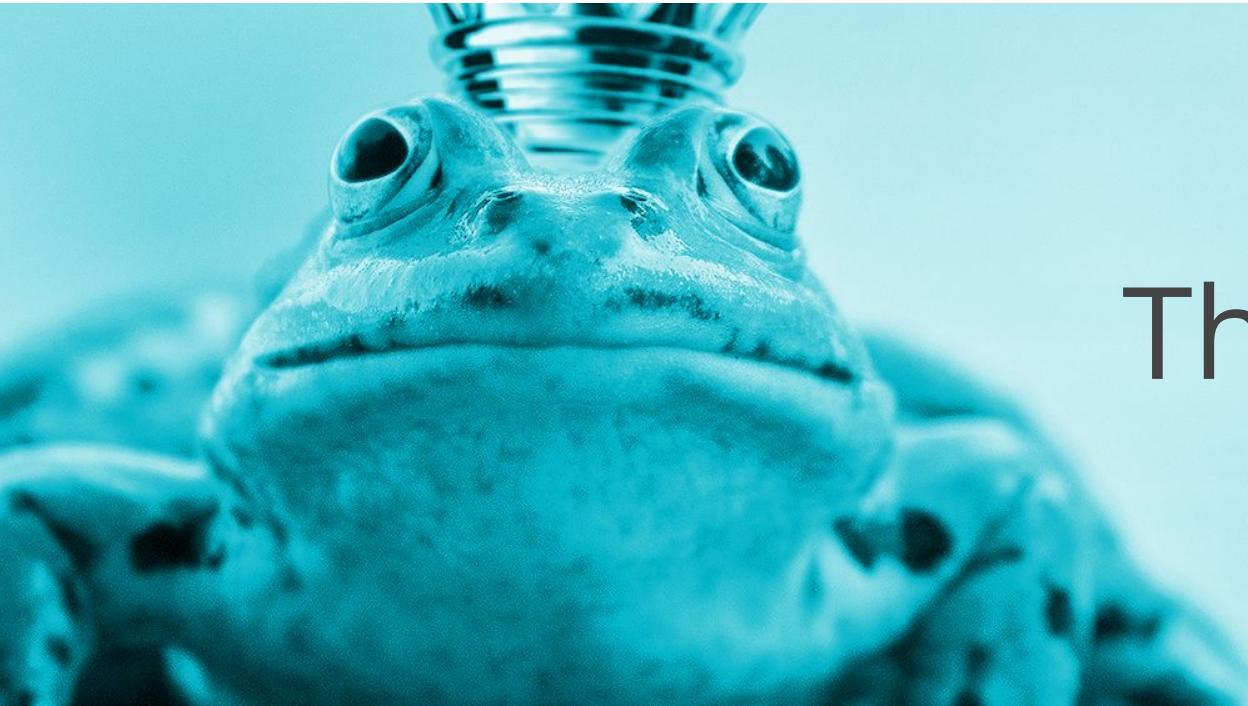


@pasqlisena

pasquale.lisena@eurecom.fr



SILKNOW



Thank you!



bit.ly/tansformer-semwebpro19