

Creating web pages from linked data with PHP and SPARQL

G. Falquet

2018-10-24

This micro-guide explains how to create php pages that query a sparql endpoint to build the page content.

1. Creating a basic php page

1. Create a sub-directory, say `linked-data`, in your web site directory.
2. Download the ARC2 library from <https://github.com/semsol/arc2/tarball/master> (on <https://github.com/semsol/arc2/wiki/>) into `linked-data`
3. Decompress the `.tar.gz` file. It should create a directory called `semsol-arc2-ebd59b0` (or something similar). Rename it to `semsol`
4. In the same directory create `test1.php` with the following content

```
1 <html>
2   <body>
3
4   <?php
5     include_once('semsol/ARC2.php'); /* ARC2 static class inclusion */
6
7
8     $dbpconfig = array(
9       "remote_store_endpoint" => "http://dbpedia.org/sparql",
10      );
11
12     $store = ARC2::getRemoteStore($dbpconfig);
13
14     if ($errs = $store->getErrors()) {
15       echo "<h1>getRemoteSotre error<h1>" ;
16     }
17
18     $query = '
```

```

19     PREFIX rdf:      <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
20     PREFIX rdfs:    <http://www.w3.org/2000/01/rdf-schema#>
21     select distinct ?l ?c
22         where {?c a rdfs:Class. ?c rdfs:label ?l}
23     limit 100
24 ' ;
25
26
27 $rows = $store->query($query, 'rows'); /* execute the query */
28
29 if ($errs = $store->getErrors()) {
30     echo "Query errors" ;
31     print_r($errs);
32 }
33
34 /* display the results in an HTML table */
35 echo "<table border='1'>" ;
36 foreach( $rows as $row ) { /* loop for each returned row */
37     print "<tr><td>" . $row['l'] . "</td><td>" . $row['c'] . "</td></tr>";
38 }
39 echo "</table>"
40
41 ?>
42 </body>
43 </html>

```

5. In a browser open the URL <http://your-site/linked-data/test1.php>

The result should look like this

Person	http://xmlns.com/foaf/0.1/Person
Class	http://www.w3.org/2002/07/owl#Class
DatatypeProperty	http://www.w3.org/2002/07/owl#DatatypeProperty
FunctionalProperty	http://www.w3.org/2002/07/owl#FunctionalProperty
ObjectProperty	http://www.w3.org/2002/07/owl#ObjectProperty
Ontology	http://www.w3.org/2002/07/owl#Ontology
Property	http://www.w3.org/1999/02/22-rdf-syntax-ns#Property
Label Property	http://xmlns.com/foaf/0.1/LabelProperty
...	...

Figure 1: Query results presented in a table

General structure of a query page

As we can see on the previous example, the php code to query a remote endpoint must proceed as the follows

- configure and “open” the endpoint (*config = ...; ARC2 :: getRemoteStore(config);*)
- define the query (*\$query = ...*)
- execute the query (*\$rows = store->query(query, 'rows');*)
- process (display) the results (foreach *\$rows* as ...)

When executed, a query of the form

```
select ?x1 ... ?xn where <condition>
```

yields a set of rows, each one containing values for the *xi*'s that satisfy the selection condition.

The standard way to process the results is through an iteration

```
1     foreach $rows as $r {
2         ...
3         <do something with $r>
4         ...
5     }
```

\$r is an associative array, with the following contents:

\$r['var'] is the value of the query variable *?var*.

\$r['var type'] is the value type, either 'uri' or 'literal' or 'bnode' (blank node).

\$r['var datatype'] is the value datatype if the value is a literal (empty if not a literal or if the datatype is not explicitly specified).

\$rvar lang' is the language of the literal value (empty if not a literal or if the language is not explicitly specified).