

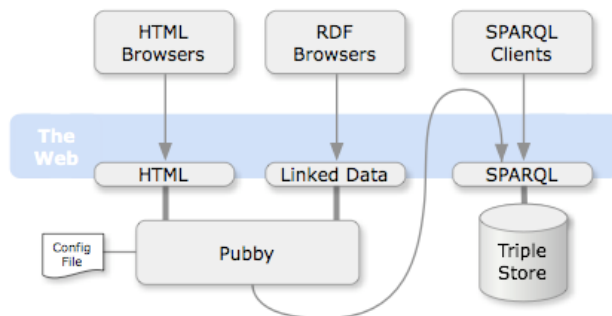
Linked Data browsers

Linked Data Browser

- One reason DBpedia is successful is its data browser for humans
- Given a URL to a Dbpedia resource, it shows all the triples with that URL as a subject or object
- A generic version of the browser software is available as [Pubby](#)
- [Elda](#) is a similar and more sophisticated system

Pubby

[Pubby](#) adds a Linked Data interface to a SPARQL endpoint



How Pubby Works

Pubby is relatively simple and does the following

- Maps published URLs an appropriate URL on the SPARQLserver
 - <http://localhost:8080/page/book1> to ??
- Sends a SPARQL describe query to get triples where the URL is either subject or object
 - Problem: Fuseki's describe returns only those where URI is subject, by default
- Converts the URLs to a prefixed form
 - <http://localhost:8080/page/book1> to ex:book1
- Displays the results as hypertext

How to set up

- Download jetty <http://download.eclipse.org/jetty/>
 - Launch by invoking `java -jar start.jar`
 - Check it it's running at <http://localhost:8080/>
- Download pubby from <http://bit.ly/Pubby>
 - `ln -s jetty-distribution-8.1.10.v20130312 jetty`
 - `ln -s pubby-0.3.3/ pubby`
- Copy pubby's webapp directory into Jetty's webapps folder and rename to root
 - `cp -R pubby/webapp jetty/webapps/root`
- Modify the configuration file to use Fuseki, etc.
- Visit <http://localhost:8080/>

Config File

```
<> a conf:Configuration;  
# Project name for display in page titles  
conf:projectName "UMBC CMSC691s13 example";  
# Homepage for the link in the page header  
conf:projectHomepage <http://bit.ly/691S13>;  
# Pubby root where the webapp is running  
conf:webBase <http://localhost:8080>;  
# URL of an RDF file with prefix mapping  
conf:usePrefixesFrom <http://umbc.edu/~finin/prefixes.n3>;  
# Preferred language for labels  
conf:defaultLanguage "en";  
# When server homepage is accessed, use this  
conf:indexResource <http://example.org/book/collection>;  
conf:dataset [  
# URL of SPARQL endpoint for dataset  
conf:sparqlEndpoint <http://localhost:3030/ds/query>;  
# Common URI prefix of resource URIs in SPARQL dataset  
conf:datasetBase <http://example.org/book>; ] .
```

An example

Property	Value
ex:has_book	<ul style="list-style-type: none">ex:book1ex:book2ex:book3ex:book4ex:book5ex:book6ex:book7

Two issues

- I had to create an external web-accessible file with prefix declarations due to a minor Pubby bug
 - may be fixed in the newest version
- Fuseki's response to a describe doesn't include triples where the URI is the object
 - You can recompile Fuseki, including an option for this
 - Or use another triple store, e.g., Sesame