A JavaScript RDF store and application library for linked data client applications

Antonio Garrote
María N. Moreno García
Motivation

Effective use of RDF as the data layer for stand-alone JS applications?
Assumptions

- Stand-alone JS applications
- RDF data
- RESTful APIs
- Read/Write support
- Integration of different data sources
- Different platforms: desktop browsers, mobile devices
- Online/Offline support
Proposed Solution

- RDFStore-JS
  - data storage
  - data query
    - [GitHub](https://github.com/antoniogarrote/rdfstore-js)
    - npm install rdfstore

- SemanticKO
  - Presentation logic
  - User interaction
    - [GitHub](https://github.com/antoniogarrote/semantic-ko)
RDFStore-js

- JS RDF storage library
- JS SPARQL query library (1.1+Update)
- Browser and Node.js support
- Evented API
- Support for different RDF serializations
- Persistence, WebWorkers
RDFStore-js: Architecture
RDFStore-js: Architecture

- Lexicon storage + B-Tree indices
- SPARQL parser + query planner

- Different backends: synchronous, asynchronous, memory, MongoDB
- Browser persistence using LocalStorage API
- WebWorkers support
- SPARQL HTTP interface
- RDF JS Interfaces API for "CONSTRUCT" queries
RDFStore-js: Evented API

- JS engines are single threaded: extensive use of events
- Evented API: register SPARQL queries on the store
- Callbacks invoked when queries result set changes
- Use of the store as a triple space / blackboard system
RDFStore-js: Performance

- Implementation of LUBM benchmark included in the source code
- Queries modified due to lack of inference support
- 1 university =~ 15MB data < 1 second
SemanticKO

- Application development library
- Use of declarative bindings between DOM tree and RDF data graph
- Built on top of RDFStore evented API
- Extension of Knockout.JS library
SemanticKO: Declarative Bindings

Data graph:

t:Lisp  rdfs:label  "Lisp".

t:John_McCarthy
  foaf:name  "John McCarthy" ;
  t:inventorOf  t:Lisp.
SemanticKO: Declarative Bindings

View template:

```html
<table id="example2">
  <tr about="[t:John_McCarthy]">
    <td data-bind="text: [foaf:name]"></td>
    <td rel="[t:inventorOf]"
        data-bind="text: [rdfs:label]"></td>
  </tr>
</table>
```
SemanticKO: Declarative Bindings

Client evaluation output:

| John McCarthy | Lisp |
SemanticKO: ViewModel

Data graph:

`t:John_McCarthy foaf:name    "John McCarthy" ;
a                    foaf:Person ;
t:inventorOf   t:Lisp .

`t:Alan_Kay          foaf:name      "Alan Kay" ;
a                    foaf:Person ;
t:inventorOf   t:Smalltalk .

`t:Lisp                   rdfs:label       "Lisp" .
`t:Smalltalk           rdfs:label       "Smalltalk" .

SemanticKO: ViewModel

ViewModel:

var viewModel =
{
    people: ko.observableArray(["t:John_McCarthy",
                                 "t:Alan_Kay" ]),
    selectedPerson: ko.observable()};
SemanticKO: ViewModel

View template:

Gurus: <select data-bind="options: people,
    value: selectedPerson">
</select>

You have chosen:
<span about="selectedPerson"
data-bind="text: [foaf:name]">
</span>

<p rel="[t:inventorOf]">
    Inventor of:<span data-bind="text: [rdfs:label]">
</span>
</p>
SemanticKO: ViewModel

Client evaluation output:

You have chosen: John McCarthy
Inventor of: Lisp

You have chosen: Alan Kay
Inventor of: Smalltalk
SemanticKO: SPARQL templates

<!-- The template -->
<script id="example5-template" type="text/html">
    {{each sko.where("{?subject a foaf:Person}\")}}()
        <li about="${$value}"
            data-bind="text: [foaf:name]">
        </li>
    {{/each}}
</script>
SemanticKO: RDF Adapter Classes

sko.Class.define("ObjectSomeValuesFrom([foaf:name])", {
    decoratedName: function() {
        return "mr. " + this.getProp("[foaf:name]");
    }
});
SemanticKO: Additional examples

http://antoniogarrote.github.com/semantic-ko/
Sample Applications
social.rdf

- Aggregations of social web data using a single WebID
- Aggregated data exposed as a RESTful API
- RDFStore used in node.js backend, WebID implementation and frontend
- Administrative front-end built using SemanticKO
social.rdf
social.rdf data visualisation test

http://antoniogarrote.com/social/stream?page=1  load older data

Select a resource

- github
  - Pushed to repo antoniogarrote/rdfstore-js commits:
    - added support for MongoDB authentication by Antonio Garrote
      (a90d031c88d850x83c85d835739f4218eb3a8d6)
      visit
  - Pushed to repo antoniogarrote/rdfstore-js commits:
    - version bump in nodejs by Antonio Garrote
      (284d3e58d90ca401ab52c300332a68de34d1d74f7)
      visit
  - twitter
    - @netlabsorg the frontend shows an associated unit but I have no idea how to extract it using the query language: http://t.co/r43ZgEM
social.rdf

- Application URL:
  http://antoniogarrote.com/social/stream

- Source code URL:
  https://github.com/antoniogarrote/social.rdf
Geek Talk

- Aggregation and visualization of data from different APIs for the members of a software project
- APIs:
  - Github
  - StackOverflow
  - Twitter
  - HackerNews
  - Google Maps
Autoloading classes in Ruby without its `autoload`

rails session helper (Is this bad)

Can't get Ruby ODBC bindings to work in 1.8.6 compiled on snow leopard
Geek Talk

rails

<table>
<thead>
<tr>
<th>Q &amp; A</th>
<th>Discussions</th>
<th>Statuses</th>
<th>People</th>
<th>Location</th>
<th>Stats &amp; Visualisations</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

dhh

- Name: David Heinemeier Hansson
- Email: david@loudthinking.com
- Company: 37signals
- Blog: http://www.loudthinking.com
- Location: Chicago, USA

wycats

- Name: Yehuda Katz
- Email: wycats@gmail.com


13719 Watchers 3051 forks 31 contributors 9 resources

2872 Followers 12 Repos

2574 Followers 121 Repos

Triples loaded: 1069
Geek Talk
Geek Talk

- Application URL:
  http://antoniogarrote.com/geektalk

- Source code URL:
  https://github.com/antoniogarrote/geektalk
Related Libraries
JSON-LD Macros

- Declarative transformations of JSON APIs into JSON-LD
- Integration with RDFStore-JS

https://github.com/antoniogarrotoe/json-ld-macros
"https://api.github.com/repos/*/*/collaborator": 
{
'$.data': {
'@ns': {'ns:default': 'gh',
'ns:replace': {'avatar_url':'foaf:depiction', 'name': 'foaf:name'}},
'@context': {'gh': 'https://api.github.com/vocabulary#',
'foaf': 'http://xmlns.com/foaf/0.1/',
'foaf:depiction': {'@type': '@id'},
'gh:url' : {'@type': '@id'}},
'@type': ['https://api.github.com/vocabulary#User',
'http://xmlns.com/foaf/0.1/Person'],
'@id': [{f:valueof': 'login'},
    {f:prefix': 'http://geektalk.com/vocabulary/geek#'}],
'@only': ['url', 'avatar_url', 'name', 'login', 'url']
}
}
LinkedVis

- Data visualization from RDF graphs
- Based on the "Grammar of Graphics"
- Extended primitives for interactive manipulation
- Data embedded into visualization

https://github.com/antoniogarrote/linkedvis
Micrograph.js

- Data layer for JSON, Microdata and RDF
- Built on top of a reduced version of RDFStore-JS
- Implicit transformation from JSON, Microdata into RDF
- MongoDB query-like JSON query language
- Automatic transformation into SPARQL

https://github.com/antoniogarrote/micrograph.js
Future Work

- Improved parser and reduced sized
- Additional features: query paths, filter functions, etc
- Integrity constraints and inference
- Server side integration
- IndexedDB persistence
- Additional backends: Redis
- Improved performance