

demo

Consuming RDF data the OOP-way

Tomasz Pluskiewicz, Karol Szczepański

Introduction

Object-oriented programming (OOP) paradigm is a prevalent method for designing computer software. In OOP data structures are represented as classes, which declare a set of named properties. This is especially important in statically typed languages, where this structure is fairly rigid and usually does not change at runtime. That structure causes friction when data in other representations needs to be accessed as objects. A well-known manifestation of this issue is the object-relational impedance mismatch, which occurs when data from relational database management systems is used in an OOP program. Developers face similar issues when using RDF data, which, even though seemingly object-oriented, has a number of properties different from typical objects in OOP sense. I'd like to show two possible ways of overcoming this problem that can be used in programs written for the .NET framework.

ORM-like solution for RDF

In a past project we have created a full-featured ORM (object-RDF mapping) library called Romantic Web as an object-oriented abstraction over RDF data. We have designed a library, which offers many features commonly found in relational ORM solutions, including .NET's Language Integrated Queries (LINQ), handling inheritance and complex data structures. It uses Rob Vesse's dotNetRDF library to access actual data backend. However our design enforces a certain way for designing the RDF store and also resulted in a fairly complex tool both in terms of usage and implementation.

More details are available at <http://romanticweb.net>

Using JSON-LD in place of an ORM

In some cases it may not be necessary to introduce a complex and limiting library to access RDF data. Still, manipulating graphs at triple level is not ideal in an object-oriented program and will likely result in unreadable code. A simple solution to that problem may be JSON-LD. Plain JSON has proved an invaluable tool for transmitting and manipulating data in many languages on all platforms. By leveraging JSON-LD `@context`, developers are empowered to create a simple bi-directional mapping between objects and RDF graphs. With minimal amount of code I have created a simple tool which uses a well-established JSON manipulation library `Newtonsoft.Json` to transform JSON-LD to and from RDF. It does not offer any advanced features but can be a lightweight alternative to the aforementioned ORM or manual work with RDF triples/quads.

Code and sample are available at <https://github.com/wikibus/JsonLD.Entities>