RDF API a SPARQL

syntaxe, API, příklady

4IZ440 Reprezentace a zpracování znalostí na WWW

Josef Petrák | me@jspetrak.name

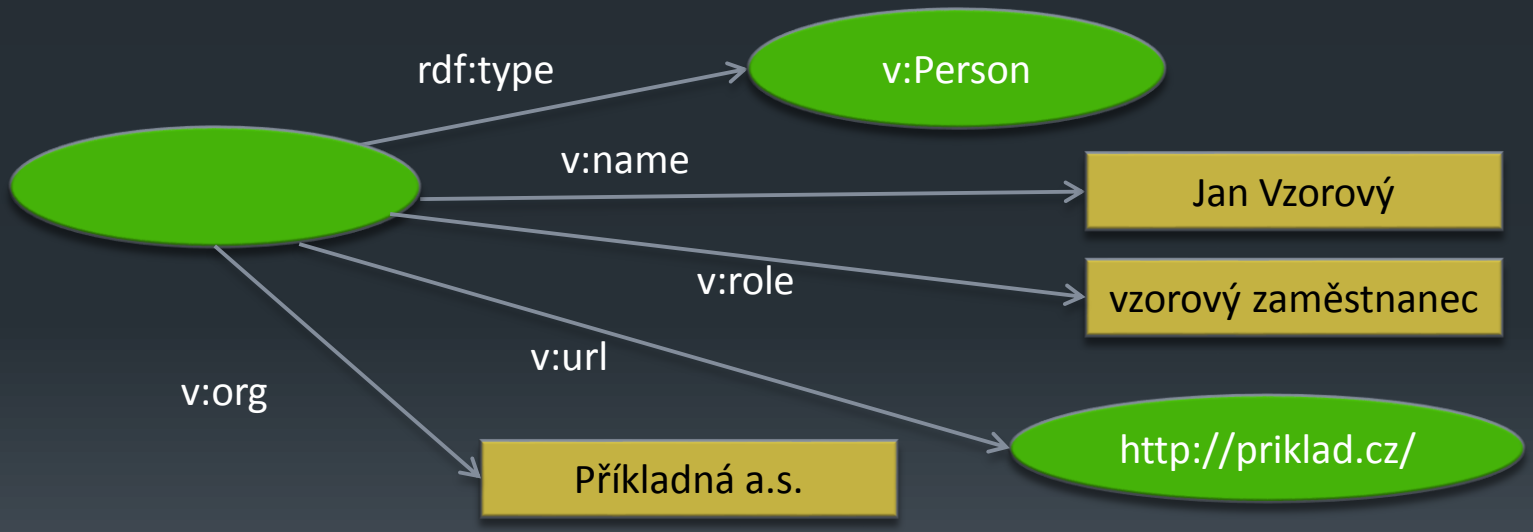
Dnes uvidíme

- Syntaxe RDF
- Základy RDF grafu
- Pojmenované grafy
- Přehled RDF API
- Příklady z Jena frameworku
- Jazyk SPARQL

Syntaxe RDF

- RDF má více syntaxí.
- Referenční syntaxí je graf – i ve formě obrazu.
- Oficiální syntaxí dle W3C je RDF/XML.
- Ostatní:
  - N-Triples
  - N3
  - RDF/JSON
  - RDFa

RDF graf



PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX v: <http://rdf.data-vocabulary.org/#>

N3 Notation

```
@prefix v: <http://rdf.data-vocabulary.org/#> .

[] a v:Person;
  v:name “Jan Vzorový”;
  v:org “Příkladná s.r.o.”;
  v:role “vzorový zaměstnanec”;
  v:url <http://priklad.cz/> .
```

RDF/XML (zkrácený zápis)

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-
ns#"
xmlns:v="http://rdf.data-vocabulary.org/#">
  <v:Person>
    <v:name>Jan Vzorový</v:name>
    <v:role>vzorový zaměstnanec</v:role>
    <v:org>Příkladná a.s.</v:org>
    <v:url rdf:resource="http://priklad.cz/"/>
  </v:Person>
</rdf:RDF>
```

N-Triples

```
_:a <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://rdf.data-vocabulary.org/#Person> .
_:a <http://rdf.data-vocabulary.org/#role> "vzorový zaměstnanec" .
_:a <http://rdf.data-vocabulary.org/#org> "Příkladná s.r.o." .
_:a <http://rdf.data-vocabulary.org/#name> "Jan Vzorový" .
_:a <http://rdf.data-vocabulary.org/#url> <http://priklad.cz/> .
```

RDF/JSON

```
{
  “_:a“ : {
    “http://www.w3.org/1999/02/22-rdf-syntax-ns#” : [
      {
        “value“ : “http://rdf.data-vocabulary.org/#Person”, “type“ : “uri“
      }
    ],
    “http://rdf.data-vocabulary.org/#name” : [
      {
        “value“ : “John Example“, “type“ : “literal“
      }
    ],
    “http://rdf.data-vocabulary.org/#affiliation” : [
      {
        “value“ : “Sample Corp. Inc.“, “type“ : “literal“
      }
    ]
  }
}
```

HTML5 + RDFa

```
<!DOCTYPE html>
<html prefix="v: http://rdf.data-vocabulary.org/#">
<head>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8" />
<title>Osobní stránka</title>
</head>
<body>
<h1>Osobní stránka</h1>
<p typeof="v:Person">
Jmenuji se <em property="v:name">Jan Vzorový</em> a pracuji v <em
property="v:role">vzorový zaměstnanec</em> pro společnost <em
property="v:org">Příkladná a.s.</em>. Moje osobní stránka: <a
href="http://priklad.cz/" rel="v:url">priklad.cz</a>.
</p>
</body>
</html>
```

Pojmenovaný graf

- Named graph (tzv. pojmenovaný graf)
- Má přiřazenou URI
- Účel? Určení původu trojic
- Čtveřice ?g ?s ?p ?o.
- TriG – rozšiřuje Turtle syntaxi o pojmenování grafu

```
<http://uri.meho.grafu/> {
  [vlastní RDF graf]
}
```

Zdroje o RDF formátech

- RDF/XML
  - http://www.w3.org/TR/rdf-syntax-grammar/
- N3 Notation
  - http://www.w3.org/DesignIssues/Notation3
  - http://www.w3.org/2000/10/swap/Primer
- N–Triples
  - http://www.w3.org/TR/rdf-testcases/#ntriples
- RDF/JSON
  - http://n2.talis.com/wiki/RDF_JSON_Specification
- RDFa
  - http://www.w3.org/TR/rdfa-syntax/
  - http://www.w3.org/TR/xhtml-rdfa-primer/

RDF API

Java

- Jena
  - http://openjena.org/
  - http://sourceforge.net/projects/jena/
- Sesame
  - http://www.openrdf.org/
- Shellac RDFa Parser
  - https://github.com/shellac/java-rdfa

PHP

- RDF API for PHP (RAP)
  - http://www4.wiwiss.fu-berlin.de/bizer/rdfapi/
  - http://sourceforge.net/projects/rdfapi-php/
- ARC2
  - http://arc.semsol.org/

Ruby

- RDF.rb
  - http://rdf.rubyforge.org/
  - https://github.com/bendiken/rdf
  - A další přídavné moduly

Modely RDF

- Model založený na trojicích
  - Pracujete s trojicemi ?predmet ?predikat ?objekt
- Model založený na zdrojích (URI)
  - Zdroje mají vlastnosti a ty hodnoty
- Model založený na ontologiích
  - Pracujete s třídami, vlastnostmi a individuy definovanými ve zvolených slovnících/schématech/ontologiích.
- Pojmenovaný graf
  - Trojice patří do grafu pojmenovaného URI adresou.
  - Pracujete s čtveřicemi ?graf ?predmet ?predikat ?objekt

Příklady ve frameworku Jena

- Práce s modelem založeným na trojicích
- Vytvoření vlastností slovníku
- Vytvoření trojic v grafu
- Vypsání trojic z grafu
- Načtení N3 souboru do grafu

Vytvoření grafu

```
package name.jspetrak;

import com.hp.hpl.jena.rdf.model.Model;
import com.hp.hpl.jena.rdf.model.ModelFactory;
import com.hp.hpl.jena.rdf.model.Property;
import com.hp.hpl.jena.rdf.model.RDFNode;
import com.hp.hpl.jena.rdf.model.Resource;
import com.hp.hpl.jena.rdf.model.Statement;
import com.hp.hpl.jena.rdf.model.StmtIterator;
import com.hp.hpl.jena.vocabulary.RDF;

public class RZZW {
    public static void main(String[] args) {
        Model model = ModelFactory.createDefaultModel();
        // Další kód sem
    }
}
```

Vytvoření vlastností slovníku

```
Property vPerson = model.createProperty("http://rdf.data-vocabulary.org/#", "Person");
Property vName = model.createProperty("http://rdf.data-vocabulary.org/#", "name");
Property vOrg = model.createProperty("http://rdf.data-vocabulary.org/#", "org");
Property vRole = model.createProperty("http://rdf.data-vocabulary.org/#", "role");
Property vUrl = model.createProperty("http://rdf.data-vocabulary.org/#", "url");
```

Vytvoření trojic grafu

```
Resource aPerson = model.createResource();

model.add(model.createStatement(aPerson, RDF.type,
vPerson));
model.add(model.createStatement(aPerson, vName, "Jan
Vzorovy"));
model.add(model.createStatement(aPerson, vOrg,
"Prikladna s.r.o."));
model.add(model.createStatement(aPerson, vRole,
"vzorovy zamestnanec"));
model.add(model.createStatement(aPerson, vUrl,
model.createResource("http://priklad.cz/")));
```

Vypsání trojic z grafu

```
StmtIterator iter = model.listStatements();

while (iter.hasNext()) {
  Statement stmt = iter.nextStatement();
  Resource  subject   = stmt.getSubject();
  Property  predicate = stmt.getPredicate();
  RDFNode object    = stmt.getObject();

  System.out.print(subject.toString());
  System.out.print(" " + predicate.toString() + " ");
  if (object instanceof Resource) {
    System.out.print(object.toString());
  } else {
    System.out.print(" \"" + object.toString() + "\"");
  }
  System.out.println(" .");
}
```

Vypsání grafu jako RDF/XML

```
model.write(System.out, "RDF/XML-ABBREV");
```

Dotazování modelu

```
StmtIterator si = model.listStatements(null, RDF.type, vPerson);
while (si.hasNext()) {
  Resource r = si.nextStatement().getSubject();
  StmtIterator sii = model.listStatements(r, vName, (RDFNode)null);
  if (sii.hasNext()) {
    System.out.println(sii.nextStatement().getObject().toString());
  }
}
```

Načtení dat z N3 souboru

```
InputStream is =
  FileManager.get().open("samples/people.n3");

Model model = ModelFactory.createDefaultModel();
RDFReader r = model.getReader("N3");
r.read(model, is, null);
is.close();
```

Dotazování pomocí SPARQL

```
String sQuery = "PREFIX v: …";
Query query = QueryFactory.create(sQuery);
QueryExecution exQuery =
  QueryExecutionFactory.create(query, model);

try {
  ResultSet result = exQuery.execSelect();
  while (result.hasNext()) {
    QuerySolution solution = result.nextSolution();
    Literal name = solution.getLiteral("name");
    // Další data ...
  }
} finally {
  exQuery.close();
}
```

Cvičení

- Vytvoření dat
  a. Zapište do souboru N3
  b. Definujte trojice pomocí API
- Načtěte data do grafu
- Vypište trojice
- Vyhledejte všechny osoby a vypište jejich jména

SPARQL

Co je SPARQL

- SPARQL Protocol and RDF Query Language
- W3C Standard od 15. 1. 2008
- „Read-only“ dotazovací jazyk pro RDF
- Založený na dalších standardech – RDF, XML, HTTP, (WSDL, SOAP)
- Konkurence pro jiné podobně určené jazyky – rdfDB, RDQL, SeRQL

Co je SPARUL

- SPARQL Update Language
- SPARQL s možností data měnit

Nástroje

- SPARQLer
  http://www.sparql.org/sparql.html
- RDF Validator/Converter
  http://www.rdfabout.com/demo/validator/
- DBPedia
  http://dbpedia.org/sparql
- Snorql
  http://dbpedia.org/snorql/

Příklad dotazu

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?url
FROM
<http://journal.dajobe.org/journal/2003/07/semblogs/bloggers.rdf>
WHERE {
  ?contributor foaf:name "Tim Berners-Lee" .
  ?contributor foaf:weblog ?url .
}
```

Co obsahuje SPARQL dotaz

- Promněnné – uvozují se znakem „?“
- FROM – klauzule identifikující zdrojová data
- WHERE – seznam trojic tvořících tzv. „graph pattern“

Typy SPARQL dotazů

- SELECT
- ASK
- DESCRIBE
- CONSTRUCT

SPARQL SELECT

- Proměnným přiřadí hodnoty a vrátí je v tabulce

```
@prefix foaf:
<http://xmlns.com/foaf/0.
1/> .
_:a foaf:name "Alice" .
_:a foaf:knows _:b .
_:a foaf:knows _:c .
_:b foaf:name "Bob" .
_:c foaf:name "Clare" .
_:c foaf:nick "CT" .
```

```
PREFIX foaf:
<http://xmlns.com/foaf/0.
1/>
SELECT ?nameX ?nameY
?nickY
WHERE {
  ?x foaf:knows ?y ;
  foaf:name ?nameX .
  ?y foaf:name ?nameY .
  OPTIONAL {
    ?y foaf:nick ?nickY .
  }
}
```

XML výsledek SPARQL SELECTu

```
<?xml version="1.0"?>
<sparql xmlns="http://www.w3.org/2005/sparql-results#">
  <head>
    <variable name="nameX"/> <variable name="nameY"/> <variable name="nickY"/>
  </head>
  <results>
    <result>
      <binding name="nameX">
        <literal>Alice</literal>
      </binding>
      <binding name="nameY">
        <literal>Bob</literal>
      </binding>
    </result>
    <result>
      <binding name="nameX">
        <literal>Alice</literal>
      </binding>
      <binding name="nameY">
        <literal>Clare</literal>
      </binding>
      <binding name="nickY">
        <literal>CT</literal>
      </binding>
    </result>
  </results>
</sparql>
```

SPARQL ASK

- Testuje, jestli graph pattern má řešení. Vrací boolean.

```
@prefix foaf:
<http://xmlns.com/foaf/0.
1/> .
_:a foaf:name "Alice" .
_:a foaf:knows _:b .
_:a foaf:knows _:c .
_:b foaf:name "Bob" .
_:c foaf:name "Clare" .
_:c foaf:nick "CT" .
```

```
PREFIX foaf:
<http://xmlns.com/foaf/0.1
/>
ASK {
  ?x foaf:name „Alice“
}
```

XML výsledek SPARQL ASKu

```
<?xml version="1.0"?>
<sparql xmlns="http://www.w3.org/2005/sparqlresults#">
  <head>
  </head>
  <results>
    <boolean>true</boolean>
  </results>
</sparql>
```

SPARQL DESCRIBE

- Vrací podgraf vyhovující graph patternu

```
DESCRIBE ?person
WHERE {
  ?person foaf:name "Tim Berners-Lee"
}
```

SPARQL DESCRIBE

```
DESCRIBE <http://example.org/>

PREFIX foaf: <http://xmlns.com/foaf/0.1/>
DESCRIBE ?x ?y <http://example.org/>
WHERE {
   ?x foaf:knows ?y
}
```

SPARQL CONSTRUCT

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix alice: <http://alice.name/#>
alice:me foaf:name "Alice" .
alice:me foaf:mbox <mailto:alice@example.org> .

PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX vcard: <http://www.w3.org/2001/vcard-rdf/3.0#>
PREFIX alice: <http://alice.name/#>
CONSTRUCT {
  <http://example.org/person#Alice> vcard:FN ?name
}
WHERE {
  alice:me foaf:name ?name .
}

@prefix vcard: <http://www.w3.org/2001/vcard-rdf/3.0#> .
<http://example.org/person#Alice> vcard:FN "Alice" .
```

SPARQL endpoint

SPARQL endpoint

- Defacto webová služba
- Identifikovaná pomocí URI
- SPARQL definuje komunikační protokol
- Příklady aktivních endpointů
  http://www.w3.org/wiki/SparqlEndpoints

Web API

- Jaký je rozdíl web API proti SPARQL endpointu?

Cvičení

- Dříve vytvořená data v N3/pomocí API dotazujte přes SPARQL
- Vypište všechny osoby, jejich firmy a role v nich

Dotazy?