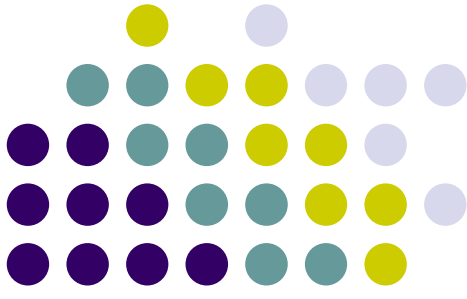


(Java) RDF APIs

Josef Petrák, Jan Zemánek
(KIZI VŠE, Praha)





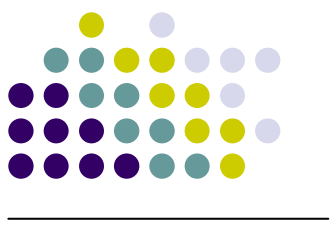
Přehled Java RDF APIs

- Jena
 - <http://jena.sourceforge.net/>
 - Download
 - <http://jena.sourceforge.net/downloads.html>
 - JavaDoc
 - <http://jena.sourceforge.net/javadoc/>
 - **An Introduction to RDF and the Jena RDF API**
 - http://jena.sourceforge.net/tutorial/RDF_API/
- NG4J (Named Graphs for Jena)
 - <http://www4.wiwi4.fu-berlin.de/bizer/ng4j/>
- Sesame
 - <http://www.openrdf.org/>
- Open Anzo
 - <http://openanzo.org/>



Přehled dalších RDF APIs

- PHP
 - RAP - RDF API for PHP
 - <http://www4.wiwiss.fu-berlin.de/bizer/rdfapi/>
 - ARC RDF Classes for PHP
 - <http://arc.semsol.org/>
- Ruby
 - ActiveRDF
 - <http://www.activerdf.org/>
- atd.
 - více v
 - **Developers Guide to Semantic Web Toolkits for different Programming Languages**
 - <http://www4.wiwiss.fu-berlin.de/bizer/toolkits/>



Modely přístupu

- **statement-centric**
 - model založený na trojicích: v grafu je sada trojic, které dotazujeme konkrétním vzorem **?s ?p ?o**.
- **resource-centric**
 - model založený na zdrojích: v grafu je sada zdrojů s vlastnostmi

Dotazování triple-based modelu



- Pro dotazování se položí vzor určité trojice a pak se prochází výsledky
- `find(null, rdf:type, keg:Person)` vyústí v sadu všech tvrzení s libovolným předmětem a výše uvedeným `p. + o.`, tj. seznamem URRef lidí v prohledávaném grafu.
- Pro složitější dotazy jsou nutné vnořené dotazy s použitím mezivýsledků jako vstupních dat.
- Příklady v článku „Parsování FOAF s PHP“
<http://dsic.zapisky.info/RDF/FOAF/parsingWithPHP/>



Triple-based model – příklad 1.

Práce s triple-based modelem v API Jena.

„Načtení RDF grafu a vypsání všech trojic.“

```
String url = „http:// ...“;  
Model model =  
    ModelFactory.createDefaultModel();  
model.read(url, null);  
  
StmtIterator i = model.listStatements();  
While (i.hasNext()) {  
    Statement s = (Statement)i.next();  
}
```



Triple-based model – příklad 2.

„Vypsání všech tvrzení o *Ing. Martinu Švihlovi*.“

```
Resource rMartinSvihla =
    model.createResource("http://keg.vse.cz/entities/person#ing-
    martin-svihla");
StmtIterator i = model.listStatements(
    new SimpleSelector(
        rMartinSvihla, null, (RDFNode) null)
    );
while (i.hasNext()) {
    Statement s = (Statement) i.next();

    Property p = s.getPredicate();
    Resource o = s.getObject();
}
```



Použití dotazovacích jazyků

- Dotazovací jazyky umožňují vybrat z RDF grafu tabulku hodnot nebo podgraf.
- Jednodušší na použití – hlavně, pokud potřebujete pouze získávat hodnoty (objekty tvrzení)
- Nejpoužívanější **SPARQL**, **SeRQL** a **RDQL**

SPARQL – příklad 1.1



„Výpis jmen lidí a jejich osobních zájmů.“

```
PREFIX keg: <http://keg.vse.cz/ns/ontology/>
SELECT ?completeName ?personalInterests
WHERE {
    ?person keg:completeName ?completeName .
    ?person keg:personalInterests
    ?personalInterests .
}
```

Jak spustit dotaz v kódu? Příklad ...



SPARQL – příklad 1.2

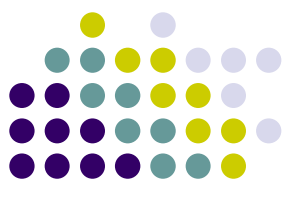
```
String sQuery = "PREFIX keg: ...";  
Query query = QueryFactory.create(sQuery);  
QueryExecution exQuery =  
    QueryExecutionFactory.create(query, model);  
  
try {  
    ResultSet result = exQuery.execSelect();  
    while (result.hasNext()) {  
        QuerySolution solution = result.nextSolution();  
        Literal completeName =  
            solution.getLiteral("completeName");  
        Literal personalInterest =  
            solution.getLiteral("personalInterests");  
    }  
} finally {  
    exQuery.close();  
}
```



SPARQL – příklad 2.

„Výpis jmen lidí s titulem ,Ing.“

```
PREFIX keg: <http://keg.vse.cz/ns/ontology/>
PREFIX rdf:
    <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT ?completeName
WHERE {
    ?person rdf:type keg:Person .
    ?person keg:completeName ?completeName .
FILTER
    regex ( ?completeName, "Ing." ) .
}
```



Samostatná úloha

Jako zdroj dat použijte soubor:

<http://zapisky.info/media/1/people-keg.rdf>

***Pomocí rozhraní Jena a triple-based modelu
napíšte aplikaci, která z výše zmíněného
RDF grafu vypíše jména těch lidí, kteří
mají více, jak jednu e-mailovou adresu.***

Tj. vlastnost `keg:email`.